



TI-Nspire™ CX

Guida di riferimento

Per sapere di più sulle tecnologie TI, consultare la guida online all'indirizzo
education.ti.com/eguide.

Informazioni importanti

Salvo per quanto dichiarato espressamente nella licenza che accompagna un programma, Texas Instruments non rilascia alcuna garanzia, esplicita o implicita, incluse fra le altre le garanzie implicite di commerciabilità e di idoneità ad uno scopo particolare, per quanto riguarda programmi o materiali cartacei e rende disponibili tali materiali esclusivamente su base "tal quale." In nessun caso Texas Instruments è responsabile verso chicchessia di danni speciali, collaterali, incidentali o consequenziali in relazione con o derivanti dall'acquisto o dall'uso di questi materiali, e la sola ed esclusiva responsabilità di Texas Instruments, indipendentemente dalla forma dell'azione, non supera l'importo indicato nella licenza per il programma. Inoltre Texas Instruments non è responsabile di rivendicazioni di alcun genere contro l'uso di questi materiali da parte di chiunque.

© 2021 Texas Instruments Incorporated

I prodotti reali possono differire leggermente dalle immagini pubblicate.

Sommario

Modelli di espressione	1
Elenco alfabetico	7
A	7
B	16
C	20
D	37
E	47
F	55
G	63
I	74
L	82
M	97
N	106
O	116
P	118
Q	126
R	129
S	145
T	165
U	179
V	179
W	180
X	183
Z	184
Simboli	191
TI-Nspire™ CX II - Comandi di Disegna	216
Programmazione grafica	216
Schermata grafica	216
Impostazioni e vista predefinite	217
Messaggi di errore della schermata grafica	218
Comandi non validi in modalità grafico	218
C	220
D	221
F	225
G	227
P	228
T	230
U	232

Elementi vuoti (nulli)	233
Scelte rapide per l'inserimento di espressioni matematiche	235
EOS™ (Equation Operating System) gerarchia	237
TI-Nspire CX II - Funzioni di programmazione di TI-Basic	239
Rientro automatico in Program Editor	239
Miglioramento dei messaggi di errore per TI-Basic	239
Costanti e valori	242
Codici di errore e messaggi	243
Codici di avvertenza e messaggi	252
Informazioni Generali	254
Guida online	254
Contattare l'assistenza TI	254
Informazioni su servizi e garanzia	254
Indice	255

Modelli di espressione

I modelli di espressione rappresentano un metodo veloce per introdurre espressioni matematiche in notazione matematica standard. Un modello, quando inserito, viene visualizzato nella riga di introduzione come tanti quadratini al posto degli elementi che si possono inserire. Un cursore indica quale elemento si può inserire.

Utilizzare i tasti freccia o premere **tab** per spostare il cursore su ciascun elemento e digitare un valore o un'espressione per esso. Premere **enter** o **ctrl enter** per calcolare l'espressione.

Modello di frazione

Tasti **ctrl** **÷**



Nota: vedere anche / (divisione), pagina 193.

Esempio:

$$\frac{12}{8 \cdot 2} \quad \frac{3}{4}$$

Modello di esponente

Tasto **^**



Nota: digitare il primo valore, premere **[**, quindi digitare l'esponente. Per riportare il cursore sulla linea di base, premere la freccia a destra (**▶**).

Nota: vedere anche ^ (potenza), pagina 194.

Esempio:

$$2^3 \quad 8$$

Modello di radice quadrata

Tasti **ctrl** **x²**



Nota: vedere anche $\sqrt()$ (radice quadrata), pagina 203.

Esempio:

$$\sqrt{4} \quad 2$$
$$\sqrt{\{9,16,4\}} \quad \{3,4,2\}$$

Modello di radice ennesima

Tasti **ctrl** **^**



Nota: vedere anche $\text{root}()$, pagina 141.

Esempio:

Modello di radice ennesima

Tasti

$$\begin{array}{ll} \sqrt[3]{8} & 2 \\ \sqrt[3]{\{8,27,15\}} & \{2,3,2.46621\} \end{array}$$

Modello di funzione esponenziale e

Tasti

e

Esponenziale naturale e elevato a potenza

Nota: vedere anche **e¹()**, pagina 47.

Esempio:

$$e^1 \quad 2.71828182846$$

Modello di log

Tasto

log

Calcola il logaritmo nella base specificata.
Per la base 10 predefinita, omettere la
base.

Esempio:

$$\log_4(2.) \quad 0.5$$

Nota: vedere anche **log()**, pagina 93.

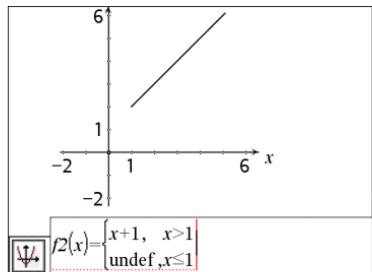
Modello di funzione piecewise a 2 tratti

Catalogo >

Consente di creare espressioni e condizioni per funzioni definite a due-tratti. Per aggiungere un tratto, fare clic sul modello e ripeterlo.

Nota: vedere anche **piecewise()**, pagina 120.

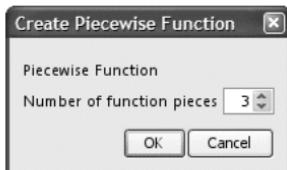
Esempio:



Modello di funzione piecewise a N tratti

Catalogo > 

Consente di creare espressioni e condizioni per una funzione definita a N -tratti. Richiede l'introduzione di N .



Esempio:

vedere l'esempio del Modello di funzione piecewise a 2 tratti.

Nota: vedere anche **piecewise()**, pagina 120.

Modello di sistema di 2 equazioni

Catalogo > 



Crea un sistema di due equazioni lineari. Per aggiungere una riga a un sistema esistente, fare clic sul modello e ripeterlo.

Nota: vedere anche **system()**, pagina 165.

Esempio:

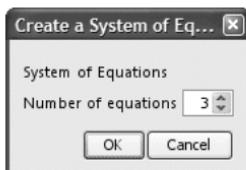
$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y\right) \quad x=\frac{5}{2} \text{ and } y=-\frac{5}{2}$$

$$\text{solve}\left(\begin{cases} y=x^2-2 \\ x+2 \cdot y=1 \end{cases}, x, y\right) \quad x=\frac{-3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1$$

Modello di sistema di N equazioni

Catalogo > 

Consente di creare un sistema di N equazioni lineari. Richiede l'introduzione di N .



Nota: vedere anche **system()**, pagina 165.

Esempio:

vedere l'esempio del Modello di sistema di equazioni (2 equazioni).

Modello di valore assoluto

Catalogo > 

 **Nota:** vedere anche **abs()**, pagina 7.

Esempio:

$$\left\{ 2,-3,4,-4^3 \right\} \quad \{ 2,3,4,64 \}$$

Modello di gg° pp'ss.ss"

Catalogo > 

 " " " "

Consente di inserire angoli nel formato **gg° pp'ss.ss"**, dove **gg** è il numero di gradi decimali, **pp** è il numero di primi e **ss.ss** è il numero di secondi.

Esempio:

$$30^{\circ}15'10" \quad 0.528011$$

Modello di matrice (2 x 2)

Catalogo > 

 [] []

Crea una matrice 2 x 2.

Esempio:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 5 \quad \begin{bmatrix} 5 & 10 \\ 15 & 20 \end{bmatrix}$$

Modello di matrice (1 x 2)

Catalogo > 

 [] .

Esempio:

$$\text{crossP}([1 \ 2], [3 \ 4]) \quad [0 \ 0 \ -2]$$

Modello di matrice (2 x 1)

Catalogo > 

 [] []

Esempio:

$$\begin{bmatrix} 5 \\ 8 \end{bmatrix} \cdot 0.01 \quad \begin{bmatrix} 0.05 \\ 0.08 \end{bmatrix}$$

Modello di matrice (m x n)

Catalogo > 

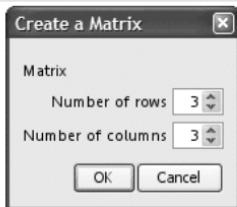
Il modello appare dopo la richiesta di specificare il numero di righe e colonne.

Esempio:

$$\text{diag} \begin{pmatrix} 4 & 2 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{pmatrix} \quad [4 \ 2 \ 9]$$

Modello di matrice (m x n)

Catalogo > 



Nota: se si crea una matrice con un numero elevato di righe e colonne, è possibile che la visualizzazione richieda un po' di tempo.

Modello di sommatoria (Σ)

Catalogo > 

$$\sum_{\square = \square}^{\square} (\square)$$

Esempio:

$$\sum_{n=3}^{7} (n) \quad 25$$

Nota: vedere anche $\Sigma()$ (**sumSeq**), pagina 205.

Modello di prodotto (\prod)

Catalogo > 

$$\prod_{\square = \square}^{\square} (\square)$$

Esempio:

$$\prod_{n=1}^{5} \left(\frac{1}{n} \right) \quad \frac{1}{120}$$

Nota: vedere anche $\prod()$ (**prodSeq**), pagina 204.

Modello di derivata prima

Catalogo > 

$$\frac{d}{dx}(\square)$$

Esempio:

$$\frac{d}{dx}(|x|)|_{x=0} \quad \text{undef}$$

Può essere utilizzato inoltre per calcolare numericamente la derivata prima in un punto con metodi di differenziazione automatica.

Modello di derivata prima

Catalogo > 

Nota: vedere anche **d()** (**derivata**), pagina 202.

Modello di derivata seconda

Catalogo > 

$$\frac{d^2}{dx^2}(\square)$$

Può essere utilizzato inoltre per calcolare numericamente la derivata seconda in un punto con metodi di differenziazione automatica.

Nota: vedere anche **d()** (**derivata**), pagina 202.

Esempio:

$$\frac{d^2}{dx^2}(x^3)|_{x=3}$$

18

Modello di integrale definito

Catalogo > 

$$\int_{\square}^{\square} \square \, d\square$$

Il modello di integrale definito può essere utilizzato per calcolare numericamente l'integrale definito, con lo stesso metodo di **nInt()**.

Nota: vedere anche **nint()**, pagina 110.

Esempio:

$$\int_0^{10} x^2 \, dx$$

333.333

ELENCO ALFABETICO

Gli elementi i cui nomi sono composti da caratteri non alfabetici (come ad esempio +, !, >) sono elencati alla fine della presente sezione, pagina 191. Se non diversamente specificato, tutti gli esempi della presente sezione sono stati eseguiti in modalità reset predefinita e tutte le variabili sono intese come non definite.

A

abs() (Valore assoluto)

Catalogo >

abs(Valore1)⇒valore

$\left \left[\frac{\pi}{2}, \frac{\pi}{3} \right] \right $	{1.5708,1.0472}
--	-----------------

abs(Lista1)⇒lista

2-3·i	3.60555
-------	---------

abs(Matrice1)⇒matrice

Restituisce il valore assoluto dell'argomento.

Nota: vedere anche **Modello di valore assoluto**, pagina 4.

Se l'argomento è un numero complesso, restituisce il modulo del numero.

Nota: tutte le variabili non definite vengono considerate come variabili reali.

amortTbl()

Catalogo >

amortTbl([NPmt,N,I,PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [valoreArrotondato])⇒matrice

amortTbl(12,60,10,5000,,12,12)

0	0.	0.	5000.
1	-41.67	-64.57	4935.43
2	-41.13	-65.11	4870.32
3	-40.59	-65.65	4804.67
4	-40.04	-66.2	4738.47
5	-39.49	-66.75	4671.72
6	-38.93	-67.31	4604.41
7	-38.37	-67.87	4536.54
8	-37.8	-68.44	4468.1
9	-37.23	-69.01	4399.09
10	-36.66	-69.58	4329.51
11	-36.08	-70.16	4259.35
12	-35.49	-70.75	4188.6

Funzione di ammortamento che restituisce una matrice come una tabella di ammortamento per un set di argomenti TVM.

NPmt è il numero di rate da includere nella tabella. La tabella inizia con la prima rata.

N, I, PV, Pmt, FV, PpY, CpY e PmtAt sono descritti nella tabella degli argomenti TVM, pagina 176.

- Se si omette Pmt, viene utilizzata l'impostazione predefinita Pmt=tvmPmt (N,I,PV,FV,PpY,CpY,PmtAt).
- Se si omette FV, viene utilizzata

- l'impostazione predefinita $FV=0$.
- Le impostazioni predefinite di PpY , CpY e $PmtAt$ sono le stesse delle funzioni TVM.

valoreArrotondato specifica il numero di cifre decimali di arrotondamento.

Impostazione predefinita=2.

Le colonne nella matrice risultante appaiono nel seguente ordine: numero di rate, interesse pagato, capitale versato e saldo.

Il saldo visualizzato nella riga n è il saldo dopo la rata n .

È possibile utilizzare la matrice di output come input per le altre funzioni di ammortamento $\Sigma\text{Int}()$ e $\Sigma\text{Prn}()$, pagina 205, e $\text{bal}()$, pagina 16.

and

Espressione booleana1 and Espressione booleana2 \Rightarrow *Espressione booleana*

Lista booleana1 and Lista booleana2 \Rightarrow *Lista booleana*

Matrice booleana1 and Matrice booleana2 \Rightarrow *Matrice booleana*

Restituisce vero o falso o una forma semplificata dell'espressione immessa originariamente.

Intero1 and Intero2 \Rightarrow *intero*

Confronta due interi reali bit per bit utilizzando un'operazione **and**.

Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati bit corrispondenti, il risultato sarà 1 se entrambi sono uguali a 1; in caso contrario il risultato sarà 0. Il valore restituito rappresenta il risultato dei bit e viene visualizzato nella modalità base che è stata impostata.

In modalità base Esadecimale:

0h7AC36 and 0h3D5F	0h2C16
--------------------	--------

Importante: è zero, non la lettera O.

In modalità base Bin:

0b100101 and 0b100	0b100
--------------------	-------

In modalità base Dec:

and

Catalogo > 

4

È possibile inserire gli interi in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli interi vengono considerati decimali (base 10).

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato.

37 and 0b100

Nota: un numero binario può contenere fino a 64 cifre (oltre al prefisso 0b). Un numero esadecimale può contenere fino ad 16 cifre.

angle() (Angolo)

Catalogo > 

angle(*Valore1*)⇒*valore*

In modalità angolo in gradi:

angle(0+2·i)

90

Restituisce l'angolo dell'argomento, interpretando l'argomento come numero complesso.

In modalità angolo in gradi (gradi centesimali):

angle(0+3·i)

100

In modalità angolo in radianti:

angle(1+i)

0.785398

angle({1+2·i,3+0·i,0-4·i})

{1.10715,0.,-1.5708}

angle(*Listal*)⇒*lista*

angle(*Matrice1*)⇒*matrice*

Restituisce una lista o una matrice di angoli degli elementi contenuti in *Listal* o *Matrice1*, interpretando ciascun elemento come un numero complesso che rappresenta un punto di coordinate rettangolari bidimensionali.

ANOVA (Analisi della varianza)

Catalogo > 

ANOVA *Lista1*,*Lista2*[,*Lista3*,...*Lista20*]
[,*Flag*]

Esegue l'analisi della varianza a una dimensione per confrontare le medie di un numero di popolazioni compreso tra due e venti. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Flag=0 per Dati, *Flag=1* per Statistiche

Variabile di output	Descrizione
stat.F	Valore della statistica F
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Grado di libertà dei gruppi
stat.SS	Somma dei quadrati dei gruppi
stat.MS	Quadrati medi dei gruppi
stat.dfError	Gradi di libertà degli errori
stat.SSError	Somma dei quadrati degli errori
stat.MSError	Quadrato medio degli errori
stat.sp	Deviazione standard aggregata
stat.xbarlist	Media dell'input delle liste
stat.CLowerList	Intervalli di confidenza al 95% per la media di ogni lista di input
stat.CUpperList	Intervalli di confidenza al 95% per la media di ogni lista di input

ANOVA2way (Analisi della varianza a due dimensioni)

ANOVA2way *Lista1,Lista2 [,Lista3,...,Lista10][,RigaLiv]*

Esegue l'analisi a due dimensioni della varianza per confrontare le medie di un numero di popolazioni compreso tra due e dieci. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

LevRow=0 per Blocco

LevRow=2,3,...,Len-1, per Due Fattori, dove Len=lunghezza(Lista1)=lunghezza(Lista2) = ... = lunghezza(Lista10) e Len / LevRow ∈ € {2,3,...}

Output: design blocco

Variabile di output	Descrizione
stat.F	Statistica F del fattore colonna
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Gradi di libertà del fattore colonna
stat.SS	Somma dei quadrati del fattore colonna
stat.MS	Quadrati medi del fattore colonna
stat.FBlock	Statistica F per fattore
stat.PValBlock	Probabilità minima in cui l'ipotesi nulla può essere rifiutata
stat.dfBlock	Gradi di libertà per fattore
stat.SSBlock	Somma dei quadrati per fattore
stat.MSBlock	Quadrati medi per fattore
stat.dfError	Gradi di libertà degli errori
stat.SSError	Somma dei quadrati degli errori
stat.MSError	Quadrati medi degli errori
stat.s	Deviazione standard dell'errore

Output FATTORE COLONNA

Variabile di output	Descrizione
stat.Fcol	Statistica F del fattore colonna
stat.PValCol	Valore di probabilità del fattore colonna
stat.dfCol	Gradi di libertà del fattore colonna
stat.SSCol	Somma dei quadrati del fattore colonna
stat.MSCol	Quadrati medi del fattore colonna

Output FATTORE RIGA

Variabile di output	Descrizione
stat.Frow	Statistica F del fattore riga
stat.PValRow	Valore di probabilità del fattore riga
stat.dfRow	Gradi di libertà del fattore riga
stat.SSRow	Somma dei quadrati del fattore riga
stat.MSRow	Quadrati medi del fattore riga

Output di INTERAZIONE

Variabile di output	Descrizione
stat.FInteract	F dell'interazione
stat.PValInteract	Valore di probabilità dell'interazione
stat.dflInteract	Gradi di libertà dell'interazione
stat.SSInteract	Somma dei quadrati dell'interazione
stat.MSInteract	Quadrati medi dell'interazione

Output di ERRORE

Variabile di output	Descrizione
stat.dfError	Gradi di libertà degli errori
stat.SSError	Somma dei quadrati degli errori
stat.MSError	Quadrati medi degli errori
s	Deviazione standard dell'errore

Ans (Ultimo risultato)

Tasti ctrl (→)

Ans⇒*valore*

Restituisce il risultato dell'ultima espressione calcolata.

56	56
56+4	60
60+4	64

approx() Approssima

Catalogo >

approx(Valore1)⇒numero

Restituisce il calcolo dell'argomento come espressione contenente valori decimali, ove possibile, indipendentemente dalla modalità corrente **Auto** o **Approssimato**.

Equivale a inserire l'argomento e a premere

[ctrl] [enter].

approx(Lista1)⇒lista

approx(Matrice1)⇒matrice

Restituisce una lista o una *matrice* nella quale ciascun elemento è stato calcolato con valori decimali, ove possibile.

approx($\frac{1}{3}$)	0.333333
approx($\left\{\frac{1}{3}, \frac{1}{9}\right\}$)	{0.333333, 0.111111}
approx({sin(π), cos(π)})	{0., -1.}
approx([sqrt(2) sqrt(3)])	[1.41421 1.73205]
approx([1 1 / 3 9])	[0.333333 0.111111]

approx({sin(π), cos(π)})	{0., -1.}
approx([sqrt(2) sqrt(3)])	[1.41421 1.73205]

►approxFraction()

Catalogo >

Valore ►approxFraction([Tol])⇒valore

Lista ►approxFraction([Tol])⇒lista

Matrice ►approxFraction([Tol])⇒matrice

Restituisce l'argomento come frazione utilizzando una tolleranza *Tol*. Se *tol* è omesso, viene utilizzata una tolleranza di 5.E-14.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **@>approxFraction(...)**.

$\frac{1}{2} + \frac{1}{3} + \tan(\pi)$	0.833333
0.8333333333333333 ►approxFraction(5.E-14)	
$\frac{5}{6}$	
{π, 1.5} ►approxFraction(5.E-14)	$\left\{ \frac{5419351}{1725033}, \frac{3}{2} \right\}$

approxRational()

Catalogo >

approxRational(Valore[, Tol])⇒valore

approxRational(Lista[, tol])⇒lista

approxRational(Matrice[, tol])⇒matrice

Restituisce l'argomento come frazione utilizzando una tolleranza *tol*. Se *tol* è omesso, viene utilizzata una tolleranza di 5.E-14.

approxRational(0.333, 5·10 ⁻⁵)	$\frac{333}{1000}$
approxRational({0.2, 0.33, 4.125}, 5.E-14)	$\left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$

arccos()**Vedere $\cos^{-1}()$, pagina 28.****arccosh()****Vedere $\cosh^{-1}()$, pagina 30.****arccot()****Vedere $\cot^{-1}()$, pagina 31.****arccoth()****Vedere $\coth^{-1}()$, pagina 31.****arccsc()****Vedere $\csc^{-1}()$, pagina 34.****arccsch()****Vedere $\csch^{-1}()$, pagina 35.****arcsec()****Vedere $\sec^{-1}()$, pagina 145.****arcsech()****Vedere $\sech^{-1}()$, pagina 146.****arcsin()****Vedere $\sin^{-1}()$, pagina 154.****arcsinh()****Vedere $\sinh^{-1}()$, pagina 156.**

augment() (Affianca/concatena)[Catalogo >](#) **augment(Lista1, Lista2)⇒lista**

augment({1,-3,2},{5,4}) {1,-3,2,5,4}

Restituisce una nuova lista in cui *Lista2* viene aggiunta (accostata) alla fine di *Lista1*.

augment(Matrice1, Matrice2)⇒matrice

Restituisce una nuova matrice in cui *Matrice2* viene aggiunta alla fine di *Matrice1*. Se si usa il carattere "," le matrici devono avere uguale numero di righe; *Matrice2* viene aggiunta a *Matrice1* come nuove colonne. Non modifica *Matrice1* o *Matrice2*.

$$\begin{array}{c|c} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1 & \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\ \hline \begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow m2 & \begin{bmatrix} 5 \\ 6 \end{bmatrix} \\ \hline \text{augment}(m1,m2) & \begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix} \end{array}$$

avgRC() (Tasso di variazione media)[Catalogo >](#) **avgRC(Espr1, Var [=Valore] [, Incr])⇒espressione**

x:=2 2

avgRC(Espr1, Var [=Valore] [, Lista1])⇒lista

avgRC(x^2-x+2,x) 3.001

avgRC(Lista1, Var [=Valore] [, Incr])⇒lista

avgRC(x^2-x+2,x,1) 3.1

avgRC(Matrice1, Var [=Valore] [, Incr])⇒matrice

avgRC(x^2-x+2,x,3) 6

Restituisce il rapporto incrementale (tasso di variazione media).

Espr1 può essere un nome di funzione definito dall'utente (vedere **Func**).

Se il *Valore* è specificato, lo stesso sostituisce qualsiasi assegnazione di variabile precedente o qualsiasi sovrapposizione corrente " | " della variabile.

Incr è il valore di incremento. Se *Incr* è omesso, viene impostato per default su 0.001.

Si noti che la funzione simile **centralDiff()** utilizza la formula del rapporto incrementale bilaterale.

B**bal()**

bal(*NPmt, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], valoreArrotondato*) \Rightarrow valore

bal(*NPmt, tabellaAmmortamento*) \Rightarrow valore

Funzione di ammortamento che calcola il saldo del piano di rientro dopo una rata specificata.

N, I, PV, Pmt, FV, PpY, CpY e PmtAt sono descritti nella tabella degli argomenti TVM, pagina 176.

NPmt specifica il numero della rata a partire dalla quale deve essere calcolato il saldo.

N, I, PV, Pmt, FV, PpY, CpY e PmtAt sono descritti nella tabella degli argomenti TVM, pagina 176.

- Se si omette *Pmt*, viene utilizzata l'impostazione predefinita *Pmt=tvmPmt (N,I,PV,FV,PpY,CpY,PmtAt)*.
- Se si omette *FV*, viene utilizzata l'impostazione predefinita *FV=0*.
- Le impostazioni predefinite di *PpY, CpY e PmtAt* sono le stesse delle funzioni TVM.

valoreArrotondato specifica il numero di cifre decimali di arrotondamento. Impostazione predefinita=2.

bal (5,6,5.75,5000,,12,12)	833.11																												
<i>tbl:=amortTbl(6,6,5.75,5000,,12,12)</i>																													
<table border="1"> <tbody> <tr> <td>0</td><td>0.</td><td>0.</td><td>5000.</td></tr> <tr> <td>1</td><td>-23.35</td><td>-825.63</td><td>4174.37</td></tr> <tr> <td>2</td><td>-19.49</td><td>-829.49</td><td>3344.88</td></tr> <tr> <td>3</td><td>-15.62</td><td>-833.36</td><td>2511.52</td></tr> <tr> <td>4</td><td>-11.73</td><td>-837.25</td><td>1674.27</td></tr> <tr> <td>5</td><td>-7.82</td><td>-841.16</td><td>833.11</td></tr> <tr> <td>6</td><td>-3.89</td><td>-845.09</td><td>-11.98</td></tr> </tbody> </table>	0	0.	0.	5000.	1	-23.35	-825.63	4174.37	2	-19.49	-829.49	3344.88	3	-15.62	-833.36	2511.52	4	-11.73	-837.25	1674.27	5	-7.82	-841.16	833.11	6	-3.89	-845.09	-11.98	
0	0.	0.	5000.																										
1	-23.35	-825.63	4174.37																										
2	-19.49	-829.49	3344.88																										
3	-15.62	-833.36	2511.52																										
4	-11.73	-837.25	1674.27																										
5	-7.82	-841.16	833.11																										
6	-3.89	-845.09	-11.98																										

bal (4, <i>tbl</i>)	1674.27
-----------------------------	---------

bal(*NPmt,tabellaAmmortamento*) calcola il saldo dopo la rata numero *NPmt* sulla base della tabella di ammortamento *tabellaAmmortamento*. L'argomento *tabellaAmmortamento* deve essere una matrice avente la forma descritta in **amortTbl()**, pagina 7.

Nota: vedere anche **ΣInt()** e **ΣPrn()**, pagina 205.

►Base2

Interol ►Base2⇒*intero*

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando **@>Base2**.

Converte *Interol* in un numero binario. I numeri binari o esadecimali hanno sempre, rispettivamente, il prefisso 0b o 0h. Zero, non la lettera O, seguito da b o h.

0b *numeroBinario*

0h *numeroEsadecimale*

Un numero binario può contenere fino a 64 cifre. Un numero esadecimale può contenere fino ad 16 cifre.

Senza prefisso, *Interol* viene considerato decimale (base 10). Il risultato viene visualizzato in modalità binaria, indipendentemente dalla modalità Base impostata.

I numeri negativi sono visualizzati nella forma “a complemento di due”. Ad esempio,

-1 è visualizzato come

0xFFFFFFFFFFFFFF in modalità base Esadecimale 0b111...111 (la cifra 1 ripetuta 64 volte) in modalità base Binaria

256►Base2	0b100000000
0h1F►Base2	0b1111

-2^{63} è visualizzato come
0h8000000000000000 in modalità base
Esadecimale e 0b100...000 (63 zeri) in
modalità base Binaria

Se viene indicato un intero decimale
esterno alla gamma di una forma binaria
con segno a 64 bit, verrà utilizzata
un'operazione a modulo simmetrico per
portare il valore nell'intervallo appropriato.
Considerare i seguenti esempi di valori
esterni alla gamma.

2^{63} diventa -2^{63} ed è visualizzato come
0h8000000000000000 in modalità base
Esadecimale e 0b100...000 (63 zeri) in
modalità base Binaria

2^{64} diventa 0 ed è visualizzato come 0h0 in
modalità base Esadecimale e 0b0 in
modalità base Binaria

$-2^{63} - 1$ diventa $2^{63} - 1$ ed è visualizzato
come 0h7FFFFFFFFFFFFF in modalità
base Esadecimale e 0b111...111 (64 - 1) in
modalità base Binaria

Interol ►Base10→*intero*

Nota: è possibile inserire questo operatore
dalla tastiera del computer digitando
@>Base10.

Converte *Interol* in numero decimale
(base 10). Le voci binarie o esadecimali
devono sempre avere, rispettivamente, il
prefisso 0b o 0h.

0b *numeroBinario*

0h *numeroEsadecimale*

Zero, non la lettera O, seguito da b o h.

0b10011►Base10	19
0h1F►Base10	31

Un numero binario può contenere fino a 64 cifre. Un numero esadecimale può contenere fino ad 8 cifre.

Senza prefisso, *Interol* viene considerato decimale (base10). Il risultato viene visualizzato in modalità decimale, indipendentemente dalla modalità Base impostata.

►Base16

Interol ►Base16⇒*intero*

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Base16.

Converte *Interol* in un numero esadecimale. I numeri binari o esadecimali hanno sempre, rispettivamente, il prefisso 0b o 0h.

0b *numeroBinario*

0h *numeroEsadecimale*

Zero, non la lettera O, seguito da b o h.

Un numero binario può contenere fino a 64 cifre. Un numero esadecimale può contenere fino ad 16 cifre.

Senza prefisso, *Interol* viene considerato decimale (base 10). Il risultato viene visualizzato in modalità esadecimale, indipendentemente dalla modalità Base impostata.

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Per ulteriori informazioni, vedere ►Base2, pagina 17.

256►Base16	0h100
0b111100001111►Base16	0hF0F

binomCdf() (Funzione della probabilità cumulativa per la distribuzione binomiale)

Catalogo > 

binomCdf(*n,p*)⇒*lista*

binomCdf(*n,p,valoreInferiore, valoreSuperiore*)⇒*numero* se *valoreInferiore* e *valoreSuperiore* sono numeri, *lista* se *valoreInferiore* e *valoreSuperiore* sono liste

binomCdf(*n,p,valoreSuperiore*) per $P(0 \leq X \leq \text{valoreSuperiore})$ ⇒*numero* se *valoreSuperiore* è un numero, *lista* se *valoreSuperiore* è una lista

Calcola la probabilità cumulativa per la distribuzione binomiale discreta con il numero di prove *n* e la probabilità di successo *p* per ciascuna prova.

Per $P(X \leq \text{valoreSuperiore})$, impostare *valoreInferiore*=0

binomPdf()

Catalogo > 

binomPdf(*n,p*)⇒*lista*

binomPdf(*n,p,ValX*)⇒*numero* se *ValX* è un numero, *lista* se *ValX* è una lista

Calcola una probabilità in corrispondenza di *ValX* per la distribuzione binomiale discreta con il numero *n* di prove e la probabilità *p* di successo per ogni prova.

C

ceiling() (Arrotondato per eccesso)

Catalogo > 

ceiling(*ValoreI*)⇒*valore*

ceiling(*.456*)

1.

Restituisce il più vicino numero intero ≤ all'argomento.

L'argomento può essere un numero reale o complesso.

Nota: vedere anche **floor()**.

ceiling() (Arrotondato per eccesso)**Catalogo > ****ceiling(Lista1)⇒lista****ceiling(Matrice1)⇒matrice**

Restituisce una lista o matrice del valore arrotondato per eccesso di ciascun elemento.

$\text{ceiling}(\{-3.1, 1.2, 5\})$	{ -3., 1.3, }
$\text{ceiling}\begin{bmatrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{bmatrix}$	$\begin{bmatrix} 0 & -3 \cdot i \\ 2. & 4 \end{bmatrix}$

centralDiff()**Catalogo > ****centralDiff(Espr1,Var [=Valore]
[,Incr])⇒espressione****centralDiff(Espr1,Var
[,Incr]) | Var=Valore⇒espressione****centralDiff(Espr1,Var [=Valore]
[,Lista])⇒lista****centralDiff(Lista1,Var [=Valore]
[,Incr])⇒lista****centralDiff(Matrice1,Var [=Valore]
[,Incr])⇒matrice**

Restituisce la derivata numerica utilizzando la formula del rapporto incrementale bilaterale.

Se il *Valore* è specificato, lo stesso sostituisce qualsiasi assegnazione di variabile precedente o qualsiasi sovrapposizione corrente “|” della variabile.

Incr è il valore di incremento. Se *Incr* è omesso, viene impostato per default su 0.001.

Se si utilizza *Lista1* o *Matrice1*, l'operazione viene mappata sui valori della lista o sugli elementi della matrice.

Nota: vedere anche .

char() (Stringa di caratteri)**Catalogo > ****char(Intero)⇒carattere**

char(38)	"&"
char(65)	"A"

char() (Stringa di caratteri)

Catalogo > 

Restituisce un carattere stringa corrispondente al numero *Intero* del set di caratteri del palmare. L'intervallo valido per intero *Intero* è compreso tra 0 e 65535.

χ^2 way

Catalogo > 

χ^2 way *MatriceOss*

chi2way *MatriceOss*

Esegue una verifica c^2 per l'associazione di numeri nella tabella a due variabili nella matrice osservata *MatriceOss*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Per informazioni sull'effetto di elementi vuoti in una matrice, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
<i>stat.chi2</i>	Statistica Chi quadrato: somma (osservati - attesi) ² /attesi
<i>stat.PVal</i>	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
<i>stat.df</i>	Gradi di libertà per le statistiche chi quadrato
<i>stat.ExpMat</i>	Matrice della tabella di numeri elementari attesi, assumendo l'ipotesi nulla
<i>stat.CompMat</i>	Matrice di contributi statistici chi quadrato elementari

χ^2 Cdf()

Catalogo > 

χ^2 Cdf

(

valoreInferiore

,*valoreSuperiore,gl*) \Rightarrow numero se

valoreInferiore e *valoreSuperiore* sono numeri, lista se *valoreInferiore* e *valoreSuperiore* sono liste

chi2Cdf

(

valoreInferiore

,*valoreSuperiore,gl*) \Rightarrow numero se
valoreInferiore e *valoreSuperiore* sono
 numeri, *list* se *valoreInferiore* e
valoreSuperiore sono liste

Calcola la probabilità della distribuzione χ^2
 tra il *valoreInferiore* e il *valoreSuperiore*
 per i gradi di libertà *gl* specificati.

Per $P(X \leq \text{valoreSuperiore})$, impostare
valoreInferiore= 0.

Per informazioni sull'effetto di elementi
 vuoti in una lista, vedere "Elementi vuoti
 (nulli)", pagina 233.

$\chi^2\text{GOF}$ *listaOss,listaAtt,gl*

chi2GOF *listaOss,listaAtt,gl*

Esegue una verifica per confermare che i
 dati del campione appartengono a una
 popolazione conforme a una data
 distribuzione. *listaOss* è una lista di
 conteggi e deve contenere numeri interi.
 Il riepilogo dei risultati è memorizzato nella
 variabile *stat.results*. (pagina 160).

Per informazioni sull'effetto di elementi
 vuoti in una lista, vedere "Elementi vuoti
 (nulli)", pagina 233.

Variabile di output	Descrizione
<i>stat.</i> χ^2	Statistica Chi quadrato: sum((osservati - attesi) ² /attesi)
<i>stat.PVal</i>	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
<i>stat.df</i>	Gradi di libertà per le statistiche chi quadrato
<i>stat.CompList</i>	Contributi statistici chi quadrato elementari

χ^2 Pdf(*ValX,gl*)⇒numero se *ValX* è un numero, lista se *ValX* è una lista

chi2Pdf(*ValX,gl*)⇒numero se *ValX* è un numero, lista se *ValX* è una lista

Calcola la funzione della densità di probabilità (pdf) per la distribuzione χ^2 a un dato valore *ValX* per i gradi di libertà *gl* specificati.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

ClearAZ (Cancella AZ)

ClearAZ

Cancella tutte le variabili con il nome di un solo carattere nello spazio attività corrente.

Se una o più variabili sono bloccate, questo comando visualizza un messaggio di errore ed elimina solo le variabili sbloccate.

Vedere **unLock**, pagina 179.

5→ <i>b</i>	5
<i>b</i>	5
ClearAZ	<i>Done</i>
<i>b</i>	"Error: Variable is not defined"

ClrErr

ClrErr

Cancella lo stato di errore e imposta la variabile di sistema *errCode* su zero.

Per un esempio di **ClrErr**, vedere l'esempio 2 del comando **Try**, pagina 172.

L'istruzione **Else** del blocco **Try...Else...EndTry** dovrebbe utilizzare **ClrErr** o **PassErr**. Se l'errore deve essere elaborato o ignorato, utilizzare **ClrErr**. Se non si sa quale azione applicare all'errore, utilizzare **PassErr** per inviarlo al successivo blocco di gestione degli errori. Se non ci sono ulteriori blocchi di gestione degli errori **Try...Else...EndTry** in attesa di applicazione, la finestra di dialogo dell'errore viene visualizzata come normale.

Nota: vedere anche **PassErr**, pagina 119 e **Try**, pagina 172.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

colAugment() (Affianca colonna)

Catalogo >

**colAugment(*Matrice1*,
Matrice2)**⇒*matrice*

Restituisce una nuova matrice in cui *Matrice2* viene aggiunta alla fine di *Matrice1*. Le matrici devono avere uguale numero di colonne; *Matrice2* viene aggiunta a *Matrice1* come nuove colonne. Non modifica *Matrice1* o *Matrice2*.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
$\begin{bmatrix} 5 & 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 & 6 \end{bmatrix}$
colAugment(<i>m1,m2</i>)	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$

colDim() (Dimensione colonna)

Catalogo >

colDim(*Matrice*)⇒*espressione*

Restituisce il numero delle colonne contenute in *Matrice*.

colDim($\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}$)	3
--	---

Nota: vedere anche **rowDim()**.

colNorm() (Norma colonna)

Catalogo >

colNorm(*Matrice*)⇒*espressione*

Restituisce il massimo delle somme dei valori assoluti degli elementi nelle colonne di *Matrice*.

$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$
colNorm(<i>mat</i>)	9

Nota: non sono ammessi elementi non definiti di una matrice. Vedere anche **rowNorm()**.

conj() (Coniugato)

Catalogo >

conj(*Valore1*)⇒*valore*

conj($1+2\cdot i$)	$1-2\cdot i$
----------------------	--------------

conj(*Listal*)⇒*lista*

conj($\begin{bmatrix} 2 & 1-3\cdot i \\ -i & -7 \end{bmatrix}$)	$\begin{bmatrix} 2 & 1+3\cdot i \\ i & -7 \end{bmatrix}$
---	--

conj(*Matrice1*)⇒*matrice*

Restituisce il complesso coniugato dell'argomento.

Nota: tutte le variabili non definite vengono considerate come variabili reali.

constructMat() (Costruisci matrice)

Catalog >

```
constructMat
(
Espr
,
Var1
,Var2,numRighe,numColonne)⇒matrice
```

Restituisce una matrice sulla base degli argomenti.

Espr è un'espressione nelle variabili *Var1* e *Var2*. Gli elementi nella matrice risultante sono formati calcolando *Espr* per ciascun valore incrementato di *Var1* e *Var2*.

Var1 è incrementato automaticamente da **1** a *numRighe*. All'interno di ciascuna riga, *Var2* è incrementato da **1** a *numColonne*.

$$\text{constructMat}\left(\frac{1}{i+j}, i, j, 3, 4\right) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 1 \\ 3 & 4 & 5 & 6 \\ 1 & 1 & 1 & 1 \\ 4 & 5 & 6 & 7 \end{bmatrix}$$

CopyVar (Copia variabile)

Catalog >

CopyVar *Var1*, *Var2*

CopyVar *Var1.*, *Var2*.

CopyVar *Var1*, *Var2* copia il valore della variabile *Var1* nella variabile *Var2*, creando *Var2* se necessario. La variabile *Var1* deve contenere un valore.

Se *Var1* è il nome di una funzione esistente definita dall'utente, copia la definizione di quella funzione nella funzione *Var2*. La funzione *Var1* deve essere definita.

Var1 deve soddisfare i requisiti validi per i nomi di variabile oppure deve essere un'espressione indiretta che viene semplificata in un nome di variabile che soddisfa i suddetti requisiti.

Define $a(x)=\frac{1}{x}$	Done
Define $b(x)=x^2$	Done
CopyVar <i>a,c:c(4)</i>	$\frac{1}{4}$
CopyVar <i>b,c:c(4)</i>	16

CopyVar (Copia variabile)

Catalogo >

CopyVar *Var1.*, *Var2.* copia tutti i membri del gruppo di variabili *Var1.* nel gruppo *Var2.*, creando *Var2.* se necessario.

Var1. deve essere il nome di un gruppo di variabili esistente, come ad esempio i risultati statistici *stat.nn* o le variabili create utilizzando la funzione **LibShortcut()**. Se *Var2.* esiste già, questo comando sostituisce tutti i termini che sono comuni a entrambi i gruppi e aggiunge i termini che non esistono ancora. Se uno o più termini di *Var2.* sono bloccati, tutti i termini di *Var2.* vengono lasciati invariati

<i>aa.a:=45</i>	45				
<i>aa.b:=6.78</i>	6.78				
CopyVar <i>aa.,bb.</i>	<i>Done</i>				
getVarInfo()	<table border="1"> <tr> <td><i>aa.a</i> "NUM" "□" 0</td> </tr> <tr> <td><i>aa.b</i> "NUM" "□" 0</td> </tr> <tr> <td><i>bb.a</i> "NUM" "□" 0</td> </tr> <tr> <td><i>bb.b</i> "NUM" "□" 0</td> </tr> </table>	<i>aa.a</i> "NUM" "□" 0	<i>aa.b</i> "NUM" "□" 0	<i>bb.a</i> "NUM" "□" 0	<i>bb.b</i> "NUM" "□" 0
<i>aa.a</i> "NUM" "□" 0					
<i>aa.b</i> "NUM" "□" 0					
<i>bb.a</i> "NUM" "□" 0					
<i>bb.b</i> "NUM" "□" 0					

corrMat() (Matrice di correlazione)

Catalogo >

corrMat(*Listal*,*List2*[...,[*Listal**List2*...*Listal*]]])

Calcola la matrice di correlazione per la matrice affiancata [*Listal* *List2* ... *Listal*].

cos() (Coseno)

Tasto

cos(*Valore1*)⇒*valore*

In modalità angolo in gradi:

$\cos\left(\frac{\pi}{4}\right)$	0.707107
$\cos(45)$	0.707107
$\cos(\{0,60,90\})$	{1.,0.5,0.}

cos(*Listal*)⇒*lista*

cos(*Valore1*) restituisce sotto forma di valore il coseno dell'argomento.

cos(*Listal*) restituisce una lista dei coseni di tutti gli elementi di *Listal*.

In modalità angolo in gradi (gradi centesimali):

$\cos(\{0,50,100\})$	{1.,0.707107,0.}
----------------------	------------------

Nota: conformemente alla modalità di misurazione degli angoli impostata, l'argomento viene interpretato come angolo in gradi, gradianti o radianti. È possibile utilizzare °, G o r per escludere provvisoriamente la modalità d'angolo selezionata.

In modalità angolo in radianti:

$\cos\left(\frac{\pi}{4}\right)$	0.707107
$\cos(45^\circ)$	0.707107

cos

In modalità angolo in radianti:

cos() (Coseno)

Tasto 

(*matriceQuadrata1*) \Rightarrow *matriceQuadrata*

Restituisce il coseno della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il coseno di ogni elemento.

Quando una funzione scalare $f(A)$ opera su *matriceQuadrata1* (A), il risultato viene calcolato dall'algoritmo:

Calcola gli autovalori (λ_i) e gli autovettori (V_i) di A .

matriceQuadrata1 deve essere diagonalizzabile. Inoltre, non può avere variabili simboliche alle quali non sia stato assegnato un valore.

Forma le matrici:

$$B = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \text{ and } X = [V_1, V_2, \dots, V_n]$$

Quindi $A = X B X^{-1}$ e $f(A) = X f(B) X^{-1}$. Ad esempio, $\cos(A) = X \cos(B) X^{-1}$ dove:

$\cos(B) =$

$$\begin{bmatrix} \cos(\lambda_1) & 0 & \dots & 0 \\ 0 & \cos(\lambda_2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \cos(\lambda_n) \end{bmatrix}$$

Tutti i calcoli vengono eseguiti in virgola mobile.

cos⁻¹() (Arcocoseno)

Tasto 

$\cos^{-1}(Valore1) \Rightarrow valore$

In modalità angolo in gradi:

$\cos^{-1}(Listal) \Rightarrow lista$

$\cos^{-1}(1)$

0.

$\cos^{-1}(Valore1)$ restituisce l'angolo il cui coseno è *Valore1*.

$\cos^{-1}(Listal)$ restituisce la lista dell'inversa dei coseni di ciascun elemento di *Listal*.

In modalità angolo in gradi (gradi centesimali):

$\cos^{-1}(0)$

100.

cos⁻¹() (Arcocoseno)

Tasto 

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradianti o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccos (...)**.

cos⁻¹
 $(matriceQuadrata1) \Rightarrow matriceQuadrata$

Restituisce il coseno inverso della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il coseno inverso di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti:

$\cos^{-1}\{0,0,2,0,5\}$

{1.5708, 1.36944, 1.0472}

In modalità angolo in radianti e in modalità formato rettangolare complesso:

$\cos^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$

$\begin{bmatrix} 1.73485+0.064606 \cdot i & -1.49086+2.10514 \\ -0.725533+1.51594 \cdot i & 0.623491+0.77836 \cdot i \\ -2.08316+2.63205 \cdot i & 1.79018-1.27182 \end{bmatrix}$

Per vedere l'intero risultato, premere ▲, quindi utilizzare ▲ e ▶ per spostare il cursore.

cosh() (Coseno iperbolico)

Catalogo > 

cosh(Valore1) \Rightarrow valore

In modalità angolo in gradi:

$\cosh \left(\left(\frac{\pi}{4} \right) r \right)$

1.74671E19

cosh(Lista1) \Rightarrow lista

cosh(Valore1) restituisce il coseno iperbolico dell'argomento.

cosh(Lista1) restituisce una lista dei coseni iperbolici di ciascun elemento di *Lista1*.

cosh
 $(matriceQuadrata1) \Rightarrow matriceQuadrata$

In modalità angolo in radianti:

$\cosh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$

$\begin{bmatrix} 421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628 \end{bmatrix}$

Restituisce il coseno iperbolico della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il coseno iperbolico di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

cosh⁻¹⁽⁾ (Arcocoseno iperbolico)

Catalogo >

cosh⁻¹(Valore1)⇒valore

cosh⁻¹(1) 0

cosh⁻¹(Lista1)⇒lista

cosh⁻¹({1,2,1,3}) {0,1.37286,1.76275}

cosh⁻¹(Valore1) restituisce l'inversa del coseno iperbolico dell'argomento.

cosh⁻¹(Lista1) restituisce una lista dell'inversa dei coseni iperbolici di ciascun elemento di *Lista1*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccosh(...)**.

cosh⁻¹

(matriceQuadrata1)⇒matriceQuadrata

Restituisce l'inversa del coseno iperbolico della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare l'inversa del coseno iperbolico di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti e in modalità formato rettangolare complesso:

$$\begin{aligned} \cosh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \\ \begin{bmatrix} 2.52503+1.73485 \cdot i & -0.009241-1.49086 \\ 0.486969-0.725533 \cdot i & 1.66262+0.623491 \\ -0.322354-2.08316 \cdot i & 1.26707+1.79018 \end{bmatrix} \end{aligned}$$

Per vedere l'intero risultato, premere **▲**, quindi utilizzare **◀** e **▶** per spostare il cursore.

cot() (Cotangente)

Tasto

cot(Valore1) ⇒ valore

In modalità angolo in gradi:

cot(Lista1) ⇒ lista

cot(45) 1.

Restituisce la cotangente di *Valore1* oppure restituisce una lista delle cotangenti di tutti gli elementi di *Lista1*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, l'argomento viene interpretato come angolo in gradi, gradi o radianti. È possibile utilizzare **°**, **G** o **r** per escludere provvisoriamente la modalità d'angolo selezionata.

In modalità angolo in gradi (gradi centesimali):

cot(50) 1.

In modalità angolo in radianti:

cot({1,2,1,3}) {0.642093,-0.584848,-7.01525}

cot⁻¹() (Arcocotangente)

Tasto 

cot⁻¹(Valore1)⇒valore

cot⁻¹(Listal)⇒lista

Restituisce l'angolo la cui cotangente è *Valore1* oppure restituisce una lista contenente l'inversa delle cotangenti di ciascun elemento di *Listal*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradi centesimali o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccot(...)**.

In modalità angolo in gradi:

cot⁻¹(1)

45.

In modalità angolo in gradi centesimali:

cot⁻¹(1)

50.

In modalità angolo in radianti:

cot⁻¹(1)

0.785398

coth() (Cotangente iperbolica)

Catalogo > 

coth(Valore1)⇒valore

coth(Listal)⇒lista

Restituisce la cotangente iperbolica di *Valore1* o restituisce una lista delle cotangenti iperboliche di tutti gli elementi di *Listal*.

coth(1.2)

1.19954

coth({1,3,2})

{1.31304,1.00333}

coth⁻¹() Arcocotangente iperbolica

Catalogo > 

coth⁻¹(Valore1)⇒valore

coth⁻¹(3.5)

0.293893

coth⁻¹(Listal)⇒lista

coth⁻¹({-2,2,1,6})

{-0.549306,0.518046,0.168236}

Restituisce la cotangente iperbolica inversa di *Valore1* oppure restituisce una lista contenente le cotangenti iperboliche inverse di ciascun elemento di *Listal*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccoth(...)**.

count()

count(Valore1oLista1 [,Valore2oLista2 ,...])⇒valore

Restituisce il totale accumulato di tutti gli elementi negli argomenti che danno come risultato valori numerici.

Gli argomenti possono essere un'espressione, un valore, una lista o una matrice. È possibile mischiare tipi di dati e utilizzare argomenti di varie dimensioni.

Per una lista, una matrice o un intervallo di celle, viene calcolato ciascun elemento per determinare se dovrebbe essere incluso nel conteggio.

In Foglio elettronico, è possibile utilizzare un intervallo di celle al posto di qualsiasi argomento.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 233.

count(2,4,6)	3
count({2,4,6})	3
count(2,{4,6},{8 10 12 14})	7

countIf()

countIf(Lista,Criteri)⇒valore

Restituisce il totale accumulato di tutti gli elementi di *Lista* che soddisfano i *Criteri* specificati.

Criteri può essere:

- Un valore, un'espressione o una stringa. Ad esempio, 3 conta solo quegli elementi di *Lista* che sono semplificati nel numero 3.
- Un'espressione booleana contenente il simbolo ? come segnaposto di ciascun elemento. Ad esempio, ?<5 conta solo quegli elementi di *Lista* che sono minori di 5.

In Foglio elettronico, è possibile utilizzare un intervallo di celle al posto di *Lista*.

Gli elementi vuoti (nulli) della lista vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 233.

countIf({1,3,"abc",undef,3,1},3)	2
----------------------------------	---

Conta il numero di elementi uguali a 3.

countIf({ "abc","def","abc",3 }, "def")	1
---	---

Conta il numero di elementi uguali a "def".

countIf({1,3,5,7,9},?<5)	2
--------------------------	---

Conta 1 e 3.

countIf({1,3,5,7,9},2<?<8)	3
----------------------------	---

Conta 3, 5 e 7.

countIf({1,3,5,7,9},?<4 or ?>6)	4
---------------------------------	---

countif()**Catalogo >**

Nota: vedere anche **sumIf()**, pagina 164 e
frequency(), pagina 60.

Conta 1, 3, 7 e 9.

cPolyRoots()**Catalogo >**

cPolyRoots(Poli,Var)⇒lista

cPolyRoots(ListaDiCoeff)⇒lista

La prima sintassi, **cPolyRoots(Poli,Var)**, restituisce una lista di radici complesse del polinomio *Poli* in funzione della variabile specificata *Var*.

Poli deve essere un polinomio in forma normale in una variabile. Non utilizzare polinomi non ridotti a forma normale, quali y^2-y+1 o $x\cdot x+2\cdot x+1$

La seconda sintassi, **cPolyRoots(ListaDiCoeff)**, restituisce una lista di radici complesse per i coefficienti di *ListadiCoeff*.

Nota: vedere anche **polyRoots()**, pagina 122.

$\text{polyRoots}(y^3+1,y)$	{-1}
$\text{cPolyRoots}(y^3+1,y)$	{-1,0.5-0.866025i,0.5+0.866025i}
$\text{polyRoots}(x^2+2\cdot x+1,x)$	{-1,-1}
$\text{cPolyRoots}(\{1,2,1\})$	{-1,-1}

crossP() (Prodotto vettoriale)**Catalogo >**

crossP(Lista1,Lista2)⇒lista

Restituisce sotto forma di lista il prodotto vettoriale di *Lista1* e *Lista2*.

Lista1 e *Lista2* devono essere uguali, 2 o 3.

crossP(Vettore1,Vettore2)⇒vettore

Restituisce un vettore riga o colonna (a seconda degli argomenti) corrispondente al prodotto vettoriale di *Vettore1* per *Vettore2*.

Vettore1 e *Vettore2* devono essere entrambi vettori riga o vettori colonna. Le dimensioni di entrambi devono essere uguali, 2 o 3.

$\text{crossP}(\{0.1,2.2,-5\},\{1,-0.5,0\})$	{-2.5,-5,-2.25}
$\text{crossP}([1 \ 2 \ 3],[4 \ 5 \ 6])$	[-3 \ 6 \ -3]
$\text{crossP}([1 \ 2],[3 \ 4])$	[0 \ 0 \ -2]

csc() (Cosecante)

Tasto 

csc(Valore1)⇒valore

csc(Lista1)⇒lista

Restituisce la cosecante di *Valore1* oppure restituisce una lista contenente le cosecanti di tutti gli elementi di *Lista1*.

In modalità angolo in gradi:

csc(45)

1.41421

In modalità angolo in gradi centesimali:

csc(50)

1.41421

In modalità angolo in radianti:

csc({1, π/2, π/3})

{1.18841, 1.1547}

csc⁻¹() (Cosecante inversa)

Tasto 

csc⁻¹(Valore1)⇒valore

csc⁻¹(Lista1)⇒lista

Restituisce l'angolo la cui cosecante è *Valore1* oppure restituisce una lista contenente le cosecanti inverse di ciascun elemento di *Lista1*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradi o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccsc (...)**.

In modalità angolo in gradi:

csc⁻¹(1)

90.

In modalità angolo in gradi centesimali:

csc⁻¹(1)

100.

In modalità angolo in radianti:

csc⁻¹({1, 4, 6})

{1.5708, 0.25268, 0.167448}

csech (Cosecante iperbolica)

Catalogo > 

csech(Valore1)⇒valore

csech(Lista1)⇒lista

Restituisce la cosecante iperbolica di *Valore1* oppure restituisce una lista di cosecanti iperboliche di tutti gli elementi di *Lista1*.

csech(3)

0.099822

csech({1, 2, 1, 4})

{0.850918, 0.248641, 0.036644}

csch⁻¹() (Cosecante iperbolica inversa)**Catalogo > ****csch⁻¹(Valore1) ⇒ valore**csch⁻¹(1) 0.881374**csch⁻¹(Listal) ⇒ lista**csch⁻¹({1,2,1,3}) {0.881374,0.459815,0.32745}

Restituisce la cosecante iperbolica inversa di *Valore1* oppure restituisce una lista contenente le cosecanze iperboliche inverse di ciascun elemento di *Listal*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccsch(...)**.

CubicReg (Regressione cubica)**Catalogo > ****CubicReg X, Y[, [Freq] [, Categoria, Includi]]**

Calcola la regressione polinomiale cubicay = a · x³+b · x²+c · x+d sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
stat.a, stat.b, stat.c, stat.d	Coefficienti di regressione
stat.R ²	Coefficiente di determinazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

cumulativeSum()

Catalogo > 

cumulativeSum(ListaI)⇒lista

cumulativeSum({1,2,3,4}) {1,3,6,10}

Restituisce una lista delle somme cumulative degli elementi in *ListaI*, incominciando dall'elemento 1.

cumulativeSum(MatriceI)⇒matrice

Restituisce una matrice delle somme cumulative degli elementi di *MatriceI*. Ciascun elemento è la somma cumulativa della colonna, dall'alto al basso.

Un elemento vuoto (nullo) in *ListaI* o *MatriceI* produce un elemento vuoto nella lista o matrice risultante. Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 233.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

cumulativeSum(m1)

$$\begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 12 \end{bmatrix}$$

Cycle (Ripeti)

Catalogo > 

Ripeti

Funzione che elenca le somme degli interi da 1 a 100 saltando 50.

Trasferisce il controllo della funzione alla iterazione immediatamente successiva del ciclo corrente (**For**, **While** o **Loop**).

Cycle non è ammesso al di fuori delle tre strutture di ciclo (**For**, **While** o **Loop**).

Cycle (Ripeti)

Catalogo >

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define <code>g()</code> =Func	<i>Done</i>
Local <code>temp,i</code>	
$0 \rightarrow temp$	
For <code>i,1,100,1</code>	
If $i=50$	
Cycle	
$temp+i \rightarrow temp$	
EndFor	
Return <code>temp</code>	
EndFunc	
 <code>g()</code>	5000

►Cylind (Forma cilindrica)

Catalogo >

Vettore ►Cylind

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Cylind.

Visualizza il vettore riga o colonna nella forma cilindrica [r,∠θ, z].

Vettore deve avere esattamente tre elementi. Può essere una riga o una colonna.

D

dbd()

Catalogo >

dbd(*data1,data2*)⇒*valore*

Restituisce il numero di giorni tra la *data1* e la *data2* usando il metodo di conteggio dei giorni effettivi.

data1 e *data2* possono essere numeri o liste di numeri all'interno di un intervallo di date del calendario normale. Se sia *data1* che *data2* sono liste, esse devono contenere lo stesso numero di elementi.

data1 e *data2* devono essere comprese tra gli anni 1950 e 2049.

<code>dbd(12.3103,1.0104)</code>	1
<code>dbd(1.0107,6.0107)</code>	151
<code>dbd(3112.03,101.04)</code>	1
<code>dbd(101.07,106.07)</code>	151

È possibile inserire le date in uno dei due formati, che differiscono esclusivamente per la posizione del punto decimale.

MM.GGAA (formata usato generalmente negli Stati Uniti)

GGMM.AA (formato usato generalmente in Europa)

►DD (Visualizza angolo decimale)

Valore1 ►DD⇒*valore*

Listal ►DD⇒*lista*

Matrice1 ►DD⇒*matrice*

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>DD.

Restituisce l'equivalente decimale dell'argomento espresso in gradi.

L'argomento è un numero, una lista o una matrice interpretata in gradi, radianti o gradi dall'impostazione della modalità Angolo.

In modalità angolo in gradi:

(1.5°)►DD	1.5°
(45°22'14.3")►DD	45.3706°
{(45°22'14.3",60°0'0")}►DD	{45.3706°,60°}

In modalità angolo in gradi centesimali:

1►DD	$\frac{9}{10}$
------	----------------

In modalità angolo in radianti:

(1.5)►DD	85.9437°
----------	----------

►Decimal (Decimale)

Valore1 ►Decimal⇒*valore*

$\frac{1}{3}$ ►Decimal	0.333333
------------------------	----------

Listal ►Decimal⇒*valore*

Matrice1 ►Decimal⇒*valore*

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Decimal.

Visualizza l'argomento nella forma decimale. Questo operatore può essere utilizzato solo alla fine della riga di introduzione.

Define Var = Espressione

Define Funzione(Param1, Param2, ...) = Espressione

Definisce la variabile *Var* o la funzione *Funzione* definita dall'utente.

Parametri, quali *Param1*, sono segnaposto per il passaggio di argomenti alla funzione. Quando si chiama una funzione definita dall'utente, occorre fornire argomenti (ad esempio, valori o variabili) corrispondenti ai parametri. Una volta chiamata, la funzione calcola *Espressione* utilizzando gli argomenti forniti.

Var e *Funzione* non possono essere il nome di una variabile di sistema né una funzione o un comando predefiniti.

Nota: questa forma di **Define** equivale all'esecuzione dell'espressione:
espressione → *Funzione*
 $(Param1, Param2)$.

Define Funzione(Param1, Param2, ...) = Func Blocco EndFunc

Define Programma(Param1, Param2, ...) = Prgm Blocco EndPrgm

In questa forma, il programma o la funzione definita dall'utente può eseguire un blocco di istruzioni multiple.

Blocco può essere una singola istruzione o una serie di istruzioni su righe separate. Inoltre *Blocco* può includere espressioni e istruzioni (quali **If**, **Then**, **Else** e **For**).

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define $g(x,y)=2 \cdot x - 3 \cdot y$	Done
$g(1,2)$	-4
$1 \rightarrow a: 2 \rightarrow b: g(a,b)$	-4
Define $h(x)=\text{when}(x<2, 2 \cdot x - 3, -2 \cdot x + 3)$	Done
$h(-3)$	-9
$h(4)$	-5

Define $g(x,y)=\text{Func}$	Done
If $x > y$ Then	
Return x	
Else	
Return y	
EndIf	
EndFunc	
$g(3,-7)$	3

Define (Definisci)

Catalogo > 

Nota: vedere anche **Define LibPriv**, pagina 40 e **Define LibPub**, pagina 40.

Define $g(x,y) = \text{Prgm}$

```
If  $x > y$  Then  
Disp  $x$ , " greater than ", $y$   
Else  
Disp  $x$ , " not greater than ", $y$   
EndIf  
EndPrgm
```

Done

$g(3,-7)$

3 greater than -7

Done

Define LibPriv (Definisci libreria privata)

Catalogo > 

Define LibPriv *Var* = *Espressione*

Define LibPriv *Funzione(Param1, Param2, ...)* = *Espressione*

Define LibPriv *Funzione(Param1, Param2, ...)* = **Func**
Blocco
EndFunc

Define LibPriv *Programma(Param1, Param2, ...)* = **Prgm**
Blocco
EndPrgm

Funziona come **Define**, eccetto che definisce una variabile, una funzione o un programma libreria privata. Funzioni e programmi privati non sono elencati nel Catalogo.

Nota: vedere anche **Define**, pagina 39 e **Define LibPub**, pagina 40.

Define LibPub (Definisci libreria pubblica)

Catalogo > 

Define LibPub *Var* = *Espressione*

Define LibPub *Funzione(Param1, Param2, ...)* = *Espressione*

Define LibPub *Funzione(Param1, Param2,*

Define LibPub (Definisci libreria pubblica)

Catalogo > 

```
...) = Func  
Blocco  
EndFunc
```

```
Define LibPub Programma(Param1,  
Param2, ...) = Prgm  
Blocco  
EndPrgm
```

Funziona come **Define**, eccetto che definisce una variabile, una funzione o un programma libreria pubblica. Funzioni e programmi pubblici vengono elencati nel Catalogo dopo che la libreria è stata salvata e aggiornata.

Nota: vedere anche **Define**, pagina 39 e **Define LibPriv**, pagina 40.

deltaList()

Vedere Δ List(), pagina 89.

DelVar

Catalog > 

DelVar Var1[, Var2] [, Var3] ...

DelVar Var.

Elimina dalla memoria la variabile o il gruppo di variabili specificato.

Se una o più variabili sono bloccate, questo comando visualizza un messaggio di errore ed elimina solo le variabili sbloccate.

Vedere **unLock**, pagina 179.

$2 \rightarrow a$	2
$(a+2)^2$	16
DelVar a	Done
$(a+2)^2$	"Error: Variable is not defined"

DelVar

Catalogo >

DelVar *Var.* elimina tutti i membri del gruppo di variabili *Var.* (come ad esempio i risultati statistici *stat.nn* o le variabili create utilizzando la funzione **LibShortcut()**). Il punto *(.)* in questa forma del comando **DelVar** ne limita l'applicabilità all'eliminazione di un gruppo di variabili; non può essere applicato alla variabile semplice *Var.*

aa.a:=45	45
aa.b:=5.67	5.67
aa.c:=78.9	78.9
getVarInfo()	$\begin{bmatrix} aa.a & \text{"NUM"} & "[]." \\ aa.b & \text{"NUM"} & "[]." \\ aa.c & \text{"NUM"} & "[]." \end{bmatrix}$
DelVar aa.	Done
getVarInfo()	"NONE"

delVoid()

Catalogo >

delVoid(*Listal*) \Rightarrow *lista*

Restituisce una lista che ha il contenuto di *Listal* con tutti gli elementi vuoti (nulli) rimossi.

Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 233.

delVoid({1,void,3})	{1,3}
---------------------	-------

det() (Determinante)

Catalogo >

det(*matriceQuadrata*,
Tolleranza) \Rightarrow *espressione*

Restituisce il determinante di *matriceQuadrata*.

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tolleranza*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tolleranza* viene ignorato.

det($\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$)	-2
$\begin{bmatrix} 1.\text{E}20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow mat1$	$\begin{bmatrix} 1.\text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$
det(<i>mat1</i>)	0
det(<i>mat1</i> ,1)	1.e20

- Se si usa **[ctrl] [enter]** oppure se si imposta la modalità **Auto o Approssimato** su **Approssimato**, i calcoli verranno eseguiti in virgola mobile.
- Se *Tolleranza* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:

det() (Determinante)

Catalogo >

$5E^{-14} \cdot \max(\dim(\text{matriceQuadrata})) \cdot \text{rowNorm}(\text{matriceQuadrata})$

diag() (Diagonale)

Catalogo >

diag(Lista)⇒matrice

diag([2 4 6])

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

diag(matriceRiga)⇒matrice

Restituisce una matrice avente i valori dell'argomento lista o matrice nella diagonale principale.

diag(matriceQuadrata)⇒matriceRiga

Restituisce una matrice riga contenente gli elementi della diagonale principale di *matriceQuadrata*.

$$\begin{array}{|c|c|c|} \hline & 4 & 6 & 8 \\ \hline & 1 & 2 & 3 \\ \hline & 5 & 7 & 9 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline & 4 & 6 & 8 \\ \hline & 1 & 2 & 3 \\ \hline & 5 & 7 & 9 \\ \hline \end{array}$$

diag(Ans)

matriceQuadrata deve essere quadrata.

dim() (Dimensione)

Catalogo >

dim(Lista)⇒intero

dim({0,1,2})

3

Restituisce le dimensioni di *Lista*.

dim(Matrice)⇒lista

Restituisce le dimensioni di Matrice nella forma di una lista a due elementi {righe, colonne}.

$$\dim \begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{pmatrix}$$

{3,2}

dim(Stringa)⇒intero

dim("Hello")

5

Restituisce il numero di caratteri contenuti nella stringa *Stringa*.

dim("Hello "&"there")

11

Disp *exprOrString1 [, exprOrString2] ...*

Visualizza gli argomenti nella cronologia di *Calculator*. Gli argomenti possono essere visualizzati in successione, separati da sottili spazi.

Questo comando è utile soprattutto in programmi e funzioni per assicurare la visualizzazione dei calcoli intermedi.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define *chars*(*start,end*)=Prgm

```
For i,start,end
Disp i, " ",char(i)
EndFor
EndPrgm
```

Done

chars(240,243)

240 ð

241 ñ

242 ö

243 ö

Done

DispAt**DispAt** *int,expr1 [,expr2 ...] ...*

DispAt consente di specificare la linea in cui l'espressione o la stringa specificata verrà visualizzata sullo schermo.

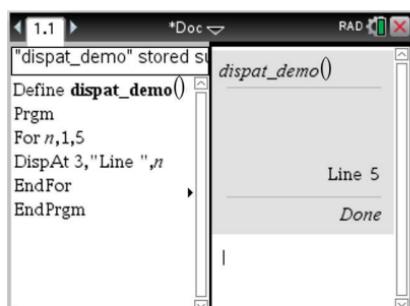
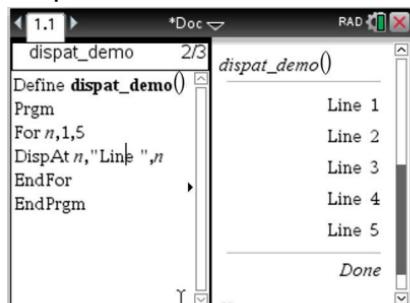
Il numero della linea può essere specificato sotto forma di espressione.

Si tenga presente che il numero della linea non è per lo schermo intero, bensì per l'area immediatamente dopo il comando/programma.

Questo comando consente un output di tipo dashboard da programmi in cui il valore di un'espressione o da una lettura del sensore viene aggiornato sulla stessa linea.

DispAt **Disp** possono essere utilizzati nello stesso programma.

Nota: il numero massimo è impostato su 8, perché corrisponde a uno schermo - pieno di linee sullo schermo del palmare - se le linee non hanno espressioni matematiche bidimensionali. Il numero esatto di linee dipende dal contenuto delle informazioni visualizzate.

DispAt**Esempio**

Esempi indicativi:

Define z()= Prgm For n,1,3 DispAt 1,"N: ",n Disp "Hello" EndFor EndPrgm	Chiusura di z() Iterazione 1: Linea 1: N:1 Linea 2: Hello Iterazione 2: Linea 1: N:2 Linea 2: Hello Linea 3: Hello Iterazione 3: Linea 1: N:3 Linea 2: Hello Linea 3: Hello Linea 4: Hello
Define z1()= Prgm For n,1,3 DispAt 1,"N: ",n EndFor For n,1,4 Disp "Hello" EndFor EndPrgm	z1() Linea 1: N:3 Linea 2: Hello Linea 3: Hello Linea 4: Hello Linea 5: Hello

Condizioni di errore:

Messaggio di errore	Descrizione
Il numero della linea DispAt deve essere compreso tra 1 e 8	L'espressione valuta il numero della linea al di fuori dell'intervallo 1-8 (inclusi)
Argomenti mancanti	Nella funzione o nel comando mancano uno o più argomenti.
Nessun argomento	Uguale alla finestra di dialogo 'errore di sintassi' corrente
Troppi argomenti	Limitare il numero di argomenti. Stesso errore di Disp.

Messaggio di errore	Descrizione
Tipo di dati non valido	Il primo argomento deve essere un numero.
Nullo: DispAt nullo	L'errore nel tipo di dati "Hello World" viene generato a vuoto (se viene definita la richiamata)

►DMS (Gradi/primi/secondi)

Catalogo > 

Valore1 ►DMS

In modalità angolo in gradi:

Lista ►DMS

$\{45.371\}$ ►DMS $45^{\circ}22'15.6''$

Matrice ►DMS

$\{\{45.371,60\}\}$ ►DMS $\{45^{\circ}22'15.6'',60^{\circ}\}$

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>DMS.

Interpreta l'argomento come un angolo e visualizza il numero DMS equivalente (GGGGGG°PP'SS.ss''). Per ulteriori informazioni sul formato DMS (gradi, primi, secondi) vedere °, ', '' pagina 210.

Nota: ►quando DMS viene utilizzato in modalità angolo in radianti, converte i radianti in gradi. Se i dati inseriti sono seguiti dal simbolo dei gradi °, non verrà eseguita alcuna conversione. ►DMS può essere utilizzato solo alla fine di una riga di introduzione.

dotP() (Prodotto scalare)

Catalogo > 

dotP(Lista1, Lista2)⇒espressione

dotP({1,2},{5,6}) 17

Restituisce il prodotto scalare di due liste.

dotP(Vettore1, Vettore2)⇒espressione

dotP([1 2 3],[4 5 6]) 32

Restituisce il prodotto scalare di due vettori.

Entrambi devono essere vettori riga o colonna.

e^() (Funzione esponenziale)**Tasto** **e^(*Valore1*)** \Rightarrow valoreRestituisce e elevato alla potenza di *Valore1*.**e¹**

2.71828

e^{3^2}

8103.08

Nota: vedere anche **e** **modello di funzione esponenziale**, pagina 2.**Nota:** premere per visualizzare e^() è diverso da accedere al carattere **E** dalla tastiera.Un numero complesso può essere inserito nella forma polare $r\text{e}^{i\theta}$. Usare questa forma solo nella modalità di misurazione degli angoli in radianti; nella modalità in gradi o gradiANTI causa un errore del dominio.**e^(*List1*)** \Rightarrow lista**e^{1,1,0,5}** { 2.71828,2.71828,1.64872 }Restituisce e elevato alla potenza di ciascun elemento di *List1*.**e^(*matriceQuadrata1*)** \Rightarrow matriceQuadrataRestituisce l'esponenziale della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare e elevato alla potenza di ciascun elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.
$$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$$
matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.**eff()****Catalogo** > **eff(*tassoNominale*,*CpY*)** \Rightarrow valore**eff(5.75,12)**

5.90398

Funzione finanziaria che converte il tasso di interesse nominale *tassoNominale* in un tasso effettivo annuo, essendo *CpY* il numero di periodi di capitalizzazione per anno.

*tassoNominal*e deve essere un numero reale e *CpY* deve essere un numero reale > 0.

Nota: vedere anche **nom()**, pagina 111.

eigVc() (A^vo^cettore)

eigVc(*matriceQuadrata*)⇒*matrice*

Restituisce una matrice contenente gli autovettori per una *matriceQuadrata* reale o complessa, in cui ogni colonna del risultato corrisponde ad un autovalore. Tenere presente che un autovettore non è univoco; esso infatti può essere scalato per qualsiasi fattore costante. Gli autovettori vengono normalizzati, cioè se $V = [x_1, x_2, \dots, x_n]$, allora:

$$x_1^2 + x_2^2 + \dots + x_n^2 = 1$$

matriceQuadrata viene dapprima equilibrata con similitudini fino a quando le norme di riga e colonna sono il più vicino possibili allo stesso valore. *matriceQuadrata* viene quindi ridotta nella forma superiore di Hessenberg mentre gli autovettori vengono calcolati con una scomposizione in fattori di Schur.

In modalità formato rettangolare complesso:

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$$

$$\begin{aligned} \text{eigVc}(m1) \\ \begin{bmatrix} -0.800906 & 0.767947 \\ 0.484029 & 0.573804+0.052258\cdot i \\ 0.352512 & 0.262687+0.096286\cdot i \end{bmatrix} \quad 0.5738 \\ 0.2626 \end{aligned}$$

Per vedere l'intero risultato, premere ▲, quindi utilizzare ▲ e ▶ per spostare il cursore.

eigVI() (Autovalore)

eigVI(*matriceQuadrata*)⇒*lista*

Restituisce la lista degli autovalori di una *matriceQuadrata* reale o complessa.

matriceQuadrata viene dapprima equilibrata con similitudini fino a quando le norme di riga e colonna sono il più vicino possibili allo stesso valore.

matriceQuadrata viene quindi ridotta nella forma superiore di Hessenberg mentre gli autovalori vengono calcolati dalla matrice superiore di Hessenberg.

In modalità formato rettangolare complesso:

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$$

$$\begin{aligned} \text{eigVI}(m1) \\ \{-4.40941, 2.20471+0.763006\cdot i, 2.20471-0\cdot i\} \end{aligned}$$

Per vedere l'intero risultato, premere ▲, quindi utilizzare ▲ e ▶ per spostare il cursore.

ElseIf**Catalogo >**

```
If Espressione booleana1 Then
  Blocco1
ElseIf Espressione booleana2 Then
  Blocco2
:
ElseIf Espressione booleanaN Then
  BloccoN
EndIf
:
```

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define $g(x) = \text{Func}$

```
If  $x \leq -5$  Then
  Return 5
ElseIf  $x > -5$  and  $x < 0$  Then
  Return  $-x$ 
ElseIf  $x \geq 0$  and  $x \neq 10$  Then
  Return  $x$ 
ElseIf  $x = 10$  Then
  Return 3
EndIf
EndFunc
```

*Done***EndFor**

Vedere For, pagina 58.

EndFunc

Vedere Func, pagina 62.

EndIf

Vedere If, pagina 74.

EndLoop

Vedere Loop, pagina 96.

EndPrgm

Vedere Prgm, pagina 124.

EndTry

Vedere Try, pagina 172.

euler ()**Catalogo >**

euler(*Espr*, *Var*, *varDipendente*, {*Var0*, *VarMax*}, *varDipendente0*, *incrVar* [, *incrEulero*]) \Rightarrow matrice

euler(*SistemaDiEspr*, *Var*, *ListaDiVarDipendenti*, {*Var0*, *VarMax*}, *ListaDiVarDipendenti0*, *incrVar* [, *incrEulero*]) \Rightarrow matrice

euler(*ListaDiEspr*, *Var*, *ListaDiVarDipendenti*, {*Var0*, *VarMax*}, *ListaDiVarDipendenti0*, *incrVar* [, *incrEulero*]) \Rightarrow matrice

Utilizza il metodo di Eulero per risolvere il sistema

$$\frac{d \text{ } depVar}{d \text{ } Var} = Expr(Var, depVar)$$

con *varDipendente*

(*Var0*)=*varDipendente0* nell'intervallo [*Var0*,*VarMax*]. Restituisce una matrice la cui prima riga definisce i valori ottenuti di *Var* e la cui seconda riga definisce il valore del primo componente della soluzione per i valori di *Var* corrispondenti, e così via.

Espr è il secondo membro che definisce l'equazione differenziale ordinaria (ODE).

SistemaDiEspr è il sistema di secondi membri che definiscono il sistema di ODE (corrisponde all'ordine di variabili dipendenti in *ListaDiVarDipendenti*).

ListaDiEspr è una lista di secondi membri che definisce il sistema di ODE (corrisponde all'ordine di variabili dipendenti in *ListaDiVarDipendenti*).

Var è la variabile indipendente.

ListaDiVarDipendenti è una lista di variabili dipendenti.

Equazione differenziale:

$$y' = 0.001 * y * (100 - y) \text{ e } y(0) = 10$$

$$\text{euler}\left\{0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right\}$$

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9 & 11.8712 & 12.9174 & 14.042 \end{bmatrix}$$

Per vedere l'intero risultato, premere \blacktriangleleft , quindi utilizzare \blacktriangleleft e \blacktriangleright per spostare il cursore.

Sistema di equazioni

$$\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

$$\text{con } y1(0) = 2 \text{ e } y2(0) = 5$$

$$\text{euler}\left\{\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0.5\}, \{2.5\}, 1\right\}$$

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. & 5. \\ 2. & 1. & 1. & 3. & 27. & 243. \\ 5. & 10. & 30. & 90. & 90. & -2070. \end{bmatrix}$$

$\{Var0, VarMax\}$ è una lista a due elementi che indica la funzione di integrare da $Var0$ a $VarMax$.

ListaDiVarDipendenti0 è una lista di valori iniziali di variabili dipendenti.

incrVar è un numero diverso da zero tale che $\text{sign}(incrVar) = \text{sign}(VarMax - Var0)$ e sono restituite soluzioni a $Var0 + i \cdot incrVar$ per tutti i valori di $i=0,1,2,\dots$ tali che $Var0 + i \cdot incrVar$ sia in $[var0, VarMax]$ (potrebbe non esserci un valore di soluzione a $VarMax$).

incrEulero è un numero intero positivo (predefinito a 1) che definisce il numero di incrementi di Eulero tra i valori ottenuti. La dimensione effettiva dell'incremento utilizzato dal metodo di Eulero è *incrVar/incrEulero*.

eval ()

eval(*Espr*) \Rightarrow stringa

eval() è valido solo in TI-Innovator™ Hub
Argomento del comando delle istruzioni di programmazione **Get**, **GetStr** e **Send**. Il software calcola l'espressione *Espr* e sostituisce l'istruzione **eval()** con il risultato come stringa di caratteri.

L'argomento *Espr* deve essere semplificato in un numero reale.

Menu Hub

Impostare l'elemento blu del LED RGB su metà intensità.

<i>lum:=127</i>	127
Send "SET COLOR.BLUE eval(lum)"	<i>Done</i>

Reimpostare l'elemento blu su OFF.

Send "SET COLOR.BLUE OFF"	<i>Done</i>
---------------------------	-------------

L'argomento **eval()** deve essere semplificato in un numero reale.

Send "SET LED eval("4") TO ON"	
"Error: Invalid data type"	

Programmare per aumentare gradualmente l'elemento rosso

```
Define fadein()=
Prgm
For i,0,255,10
  Send "SET COLOR.RED eval(i)"
  Wait 0.1
EndFor
Send "SET COLOR.RED OFF"
EndPrgm
```

Eseguire il programma.

fadein()	Done
<i>n</i> :=0.25	0.25
<i>m</i> :=8	8
<i>n· m</i>	2.
Send "SET COLOR.BLUE ON TIME eval(<i>n· m</i>)"	<i>Done</i>
<i>iostr.SendAns</i> "SET COLOR.BLUE ON TIME 2"	<i>Done</i>

Anche se **eval()** non mostra il suo risultato, è possibile visualizzare la stringa del comando Hub risultante dopo l'esecuzione del comando controllando ciascuna delle seguenti variabili speciali.

iostr.SendAns
iostr.GetAns
iostr.GetStrAns

Nota: vedere anche **Get** (pagina 64), **GetStr** (pagina 71) e **Send** (pagina 146).

Exit (Esci)

Catalogo >

Exit

Descrizione della funzione:

Permette di uscire dal blocco corrente **For**, **While** o **Loop**.

Exit non è ammesso al di fuori delle tre strutture iterative (**For**, **While** o **Loop**).

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

```
Define g()=Func
  Local temp,i
  0→temp
  For i,1,100,1
    temp+i→temp
  If temp>20 Then
    Exit
  EndIf
  EndFor
EndFunc
```

g()	21
-----	----

exp() (e alla potenza)

Tasto

exp(Valore1)⇒valore

Restituisce e elevato alla potenza di *Espr1*.

e ¹	2.71828
e ^{3²}	8103.08

exp() (e alla potenza)

Tasto 

Restituisce **e** elevato alla potenza di *Valore1*.

Nota: vedere anche **e** modello di funzione esponenziale, pagina 2.

Un numero complesso può essere inserito nella forma polare rei θ . Usare questa forma solo nella modalità di misurazione degli angoli in radianti; nella modalità in gradi o gradi causa un errore del dominio.

exp(Lista1)⇒lista

e {1,1.,0.5} {2.71828,2.71828,1.64872}

Restituisce **e** elevato alla potenza di ciascun elemento di *Lista1*.

**exp
(matriceQuadrata1)⇒matriceQuadrata**

Restituisce l'esponenziale della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare **e** elevato alla potenza di ciascun elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$	$\begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$
--	---

expr() (Da stringa a espressione)

Catalogo > 

expr(Stringa)⇒espressione

"Define cube(x)=x^3" → <i>funcstr</i>	"Define cube(x)=x^3"
expr(<i>funcstr</i>)	Done
cube(2)	8

Restituisce la stringa di caratteri contenuta in *Stringa* come espressione e la esegue subito.

ExpReg (Regressione esponenziale)

Catalogo > 

ExpReg *X*, *Y* [, *Freq*][, *Categoria*, *Includi*]]

Calcola la regressione esponenziale $y = a \cdot (b)^x$ sulle liste X e Y con frequenza $Freq$. Il riepilogo dei risultati è memorizzato nella variabile `stat.results`. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e Y sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di X e Y . Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti X e Y .

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot (b)^x$
stat.a, stat.b	Coefficienti di regressione
stat.r ²	Coefficiente di determinazione lineare di dati trasformati
stat.r	Coefficiente di correlazione per dati trasformati ($x, \ln(y)$)
stat.Resid	Residui associati al modello esponenziale
stat.ResidTrans	Residui associati all'adattamento lineare dei dati trasformati
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

factor() (Fattorizza)**Catalogo >**

factor(*numeroRazionale*) restituisce il numero razionale scomposto in fattori primi. Per i numeri composti, il tempo di elaborazione cresce in modo esponenziale secondo il numero di cifre del secondo fattore più grande. Ad esempio, la scomposizione in fattori di un intero di 30 cifre può richiedere più di un giorno, mentre la scomposizione di un numero di 100 cifre può richiedere più di un secolo.

<code>factor(152417172689)</code>	123457·1234577
<code>isPrime(152417172689)</code>	false

Per arrestare manualmente un calcolo:

- **Palmare:** Tenere premuto il tasto e premere `enter` più volte.
- **Windows®:** Tenere premuto il tasto **F12** e premere **Invio** più volte.
- **Macintosh®:** Tenere premuto il tasto **F5** e premere **Invio** più volte.
- **iPad®:** L'app mostra un prompt. È possibile continuare ad attendere o annullare.

Se si desidera soltanto determinare se un numero è primo, utilizzare `isPrime()`. Ciò risulta molto più veloce, in particolare se *numeroRazionale* non è primo e se il secondo fattore più grande ha più di cinque cifre.

FCdf() (Probabilità di distribuzione F)**Catalogo >**

```
FCdf
(
estremoInf
,estremoSup,glNumer,glDenom)⇒numero
se estremoInf e estremoSup sono numeri,
lista se estremoInf e estremoSup sono liste
```

```
FCdf
(
estremoInf
,estremoSup,glNumer,glDenom)⇒numero
se estremoInf e estremoSup sono numeri,
```

FCdf() (Probabilità di distribuzione F)

Catalogo >

lista se *estremoInf* e *estremoSup* sono liste

Calcola la probabilità di distribuzione F tra *lowBound* e *upBound* per il *dfNumero* specificato (gradi di libertà) e *dfDenom*.

Per $P(X \leq upBound)$, impostare *lowBound* = 0.

Fill (Riempì)

Catalogo >

Fill Valore, varMatrice⇒*matrice*

Sostituisce ciascun elemento della variabile *varMatrice* con *Valore*.

varMatrice deve esistere già.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow amatrix$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Fill 1.01,amatrix	Done
amatrix	$\begin{bmatrix} 1.01 & 1.01 \\ 1.01 & 1.01 \end{bmatrix}$

Fill Valore, varLista⇒*lista*

Sostituisce ciascun elemento della variabile *varLista* con *Valore*.

varLista deve esistere già.

$\{1,2,3,4,5\} \rightarrow alist$	$\{1,2,3,4,5\}$
Fill 1.01,alist	Done
alist	$\{1.01,1.01,1.01,1.01,1.01\}$

FiveNumSummary

Catalog >

FiveNumSummary *X* [,*Freq*]
[,*Categoria*,*Includi*]]

Fornisce una versione abbreviata delle statistiche a 1 variabile nella lista *X*.

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

X rappresenta una lista contenente i dati.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici di categoria dei dati corrispondenti di *X*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Un elemento vuoto (nullo) in qualsiasi lista *X*, *Freq* o *Categoria* produce un elemento vuoto corrispondente in tutte queste liste. Per ulteriori informazioni sugli elementi vuoti, vedere pagina 233

Variabile di output	Descrizione
stat.MinX	Minimo dei valori x
stat.Q ₁ X	1° quartile di x
stat.MedianX	Mediana di x
stat.Q ₃ X	3° quartile di x
stat.MaxX	Massimo dei valori x

floor() (Arrotondato per difetto)

floor(*Valore l*)⇒*intero*

floor(-2.14)

-3.

Restituisce il numero intero più grande che è ≤ all'argomento. Questa funzione è identica a **int()**.

L'argomento può essere un numero reale o complesso.

floor(*Lista l*)⇒*lista*

floor{ $\left\{\frac{3}{2}, 0, -5.3\right\}$ } {1, 0, -6.}

floor(*Matrice l*)⇒*matrice*

floor{ $\begin{bmatrix} 1.2 & 3.4 \\ 2.5 & 4.8 \end{bmatrix}$ } {1. 3.} {2. 4.}

Restituisce una lista o matrice del valore arrotondato per difetto di ciascun elemento.

Nota: vedere anche **ceiling()** e **int()**.

For *Var, Basso, Alto [, Incr]*
Blocco
EndFor

Esegue iterativamente le istruzioni di *Blocco* per ciascun valore di *Var*, da *Basso* a *Alto*, secondo incrementi pari a *Incr*.

Var non deve essere una variabile di sistema.

Incr può essere un valore positivo o negativo. Il valore predefinito è 1.

Blocco può essere una singola istruzione o una serie di istruzioni separate dal carattere ":".

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define *g()*=Func
 Local *tempsum*,*step*,*i*
 $0 \rightarrow \text{tempsum}$
 $1 \rightarrow \text{step}$
 For *i*,1,100,*step*
 $\text{tempsum} + i \rightarrow \text{tempsum}$
 EndFor
 EndFunc

Done

g()

5050

format() (Formato)

Catalogo >

format(*Valore*[,
stringaFormato])=*stringa*

Restituisce *Valore* come stringa di caratteri basata sul modello di formato.

stringaFormato è una stringa e deve essere espressa nella forma: "F[n]", "S[n]", "E[n]", "G[n][c]", dove le porzioni racchiuse tra parentesi [] sono facoltative.

F[n]: formato fisso. n rappresenta il numero di cifre da visualizzare dopo la virgola decimale.

S[n]: formato scientifico. n rappresenta il numero di cifre da visualizzare dopo la virgola decimale.

E[n]: formato tecnico. n rappresenta il numero di cifre dopo la prima cifra significativa. L'esponente è modificato secondo multipli di tre e la virgola decimale viene spostata verso destra di zero, una o due cifre.

format(1.234567,"f3")	"1.235"
format(1.234567,"s2")	"1.23e0"
format(1.234567,"e3")	"1.235e0"
format(1.234567,"g3")	"1.235"
format(1234.567,"g3")	"1,234.567"
format(1.234567,"g3,r:")	"1:235"

G[n][c]: analogo al formato fisso, separa inoltre le cifre a sinistra del separatore decimal in gruppi di tre. c specifica il carattere separatore dei gruppi; il valore predefinito è la virgola. Se c è un punto, il separatore decimale viene visualizzato come virgola.

[Rc]: tutti gli indicatori precedenti possono essere seguiti dal suffisso di radice Rc, dove c è un singolo carattere che specifica che cosa sostituire al punto della radice.

fPart() Funzione parte frazionaria

fPart(*Espr1*)⇒*espressione*

fPart(-1.234)	-0.234
---------------	--------

fPart(*Lista1*)⇒*lista*

fPart({1,-2.3,7.003})	{0,-0.3,0.003}
-----------------------	----------------

fPart(*Matrice1*)⇒*matrice*

Restituisce la parte frazionaria dell'argomento.

Per una lista o una matrice, restituisce le parti frazionarie degli elementi.

L'argomento può essere un numero reale o complesso.

FPdf() (Probabilità di distribuzione F)

FPdf(*valX,glNumer,glDenom*)⇒*numero* se *ValX* è un numero, *lista* se *ValX* è una lista

FPdf(*valX,glNumer,glDenom*)⇒*numero* se *ValX* è un numero, *lista* se *ValX* è una lista

Calcola la probabilità di distribuzione F in *valX* per il *glNumer* (gradi di libertà) e *glDenom* specificati.

freqTable►list()

**freqTable►list(Lista1,Lista
ListaInteriFreq)⇒lista**

Restituisce una lista contenente gli elementi di *Lista1* espansi secondo le frequenze in *ListaInteriFreq*. Questa funzione può essere utilizzata per costruire una tabella di frequenze per l'applicazione Dati e statistiche.

Lista1 può essere qualsiasi lista valida.

ListaInteriFreq deve avere la stessa dimensione di *Lista1* e deve contenere solo elementi interi non negativi. Ciascun elemento specifica il numero di volte che l'elemento corrispondente di *Lista1* verrà ripetuto nella lista dei risultati. Un valore zero esclude l'elemento corrispondente di *Lista1*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **freqTable@>list(...)**.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

Catalogo >

freqTable►list({1,2,3,4},{1,4,3,1})
{1,2,2,2,2,3,3,3,4}
freqTable►list({1,2,3,4},{1,4,0,1})
{1,2,2,2,2,4}

frequency()

Catalogo >

frequency(Lista1,listaContenitori)⇒lista

Restituisce una lista contenente i conteggi degli elementi di *Lista1*. I conteggi sono basati su intervalli (contenitori) definiti nell'argomento *listaContenitori*.

Se *listaContenitori* è {b(1), b(2), ..., b(n)}, gli intervalli specificati sono {?≤b(1), b(1)<?≤b(2),...,b(n-1)<?≤b(n), b(n)>?}. La lista risultante è un elemento più lungo di *listaContenitori*.

datalist:={1,2,e,3,π,4,5,6,"hello",7}
{1,2,2.71828,3,3.14159,4,5,6,"hello",7}
frequency(datalist,{2.5,4.5}) {2,4,3}

Spiegazione del risultato:

2 elementi di *Datalist* sono ≤2.5

4 elementi di *Datalist* sono > 2.5 e ≤4.5

3 elementi di *Datalist* sono >4.5

L'elemento "hello" è una stringa e non può essere collocata in alcun contenitore definito.

Ciascun elemento del risultato corrisponde al numero di elementi di *Listal* che rientrano nell'intervallo di quel contenitore. Espresso nei termini della funzione **countIf()**, il risultato è { countIf(list, ?≤b(1)), countIf(list, b(1)<?≤b(2)), ..., countIf(list, b(n-1)<?≤b(n)), countIf(list, b(n)>?) }.

Gli elementi di *Listal* che non possono essere “inseriti in un contenitore” vengono ignorati. Gli elementi vuoti (nulli) vengono ugualmente ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

In Foglio elettronico, è possibile utilizzare un intervallo di celle al posto di entrambi gli argomenti.

Nota: vedere anche **countIf()**, pagina 32.

FTest_2Samp (Verifica F su due campioni)

FTest_2Samp *Listal,Lista2[,Freq1[,Freq2 [,Ipotesi]]]*

FTest_2Samp *Listal,Lista2[,Freq1[,Freq2 [,Ipotesi]]]*

(Input lista dati)

FTest_2Samp *sx1,n1,sx2,n2[,Ipotesi]*

FTest_2Samp *sx1,n1,sx2,n2[,Ipotesi]*

(Input statistiche riepilogo)

Consente di eseguire una verifica F su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Per $H_a: \sigma_1 > \sigma_2$, impostare *Ipotesi*>0

Per $H_a: \sigma_1 \neq \sigma_2$ (default), impostare *Ipotesi*=0

Per $H_a: \sigma_1 < \sigma_2$, impostare *Ipotesi*<0

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.F	Statistica F calcolata per la sequenza di dati
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.dfNumer	Gradi di libertà del numeratore = $n_1 - 1$
stat.dfDenom	Gradi di libertà del denominatore = $n_2 - 1$
stat.sx1, stat.sx2	Deviazioni standard dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</i>
stat.x1_bar	Medie dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</i>
stat.x2_bar	
stat.n1, stat.n2	Dimensione dei campioni

Func

Func

Blocco

EndFunc

Modello per la creazione di una funzione definita dall'utente.

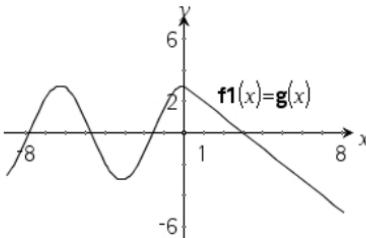
Blocco può essere una singola istruzione, una serie di istruzioni separate dal carattere ":" o una serie di istruzioni su righe separate. La funzione può utilizzare l'istruzione **Return** per restituire un dato risultato.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Definizione di una funzione Piecewise (definita a tratti):

```
Define g(x)=Func           Done
If x<0 Then
  Return 3·cos(x)
Else
  Return 3-x
EndIf
EndFunc
```

Risultato della rappresentazione grafica $g(x)$



gcd() (Massimo comun divisore)

Catalogo >

gcd(*Numero1*, *Numero2*) \Rightarrow *espressione*

gcd(18,33)

3

Restituisce il massimo comune divisore (gcd) dei due argomenti. Il gcd di due frazioni è il **massimo comune divisore** dei rispettivi numeratori diviso per il **minimo comune multiplo (lcm)** dei loro denominatori.

In modalità Auto o Approssimato, il gcd di numeri decimali in virgola mobile è 1.0.

gcd(*Lista1*, *Lista2*) \Rightarrow *lista*

gcd({12,14,16},{9,7,5}) {3,7,1}

Restituisce i massimi comuni divisori degli elementi corrispondenti in *Lista1* e *Lista2*.

gcd(*Matrice1*, *Matrice2*) \Rightarrow *matrice*

gcd[[2,4],[4,8],[6,8]] [[2,4],[12,16]]

Restituisce i massimi comuni divisori degli elementi corrispondenti in *Matrice1* e *Matrice2*.

geomCdf() (Probabilità geometrica cumulata)

Catalogo >

geomCdf

(

p

,

valoreInferiore,*valoreSuperiore*) \Rightarrow *numero*
 se *valoreInferiore* e *valoreSuperiore* sono numeri, *lista* se *valoreInferiore* e *valoreSuperiore* sono liste

geomCdf(*p*,*valoreSuperiore*) per $P(1 \leq X$ $\leq \text{valoreSuperiore}) \Rightarrow \text{numero}$ se*valoreSuperiore* è un numero, *lista* se *valoreSuperiore* è una lista

Calcola una probabilità geometrica cumulata da *valoreInferiore* a *valoreSuperiore* con la probabilità di esiti favorevoli *p* specificata.

Per $P(X \leq \text{valoreSuperiore})$, impostare *valoreSuperiore* = 1.

geomPdf(*p,ValX*)⇒*numero* se *ValX* è un numero, *lista* se *ValX* è una lista

Calcola una probabilità in corrispondenza di *ValX*, il numero della prova in cui si è verificato il primo caso favorevole, per la distribuzione geometrica discreta con la probabilità di esiti favorevoli *p* specificata.

Get**Menu Hub**

Get[*promptString*,]*var*[, *statusVar*]

Get[*promptString*,] *funk*[*arg1*, ...*argn*] [, *statusVar*]

Istruzione di programmazione: recupera un valore da uno collegato TI-Innovator™ Hub e assegna il valore alla variabile *var*.

Il valore deve essere obbligatorio:

- In anticipo, attraverso un comando **Send "LEGGI ..."**.
 - oppure —
- Incorporando una richiesta "**LEGGI ...**" come argomento *promptString* opzionale. Questo metodo consente di utilizzare un solo comando per richiedere il valore e recuperarlo.

Avviene una semplificazione隐式的. Per esempio, una stringa ricevuta di "123" viene interpretata come valore numerico. Per preservare la stringa, utilizzare **GetStr** invece di **Get**.

Se si include l'argomento opzionale *statusVar*, viene assegnato un valore basato sul successo dell'operazione. Un valore di zero significa che non è stato ricevuto nessun dato.

Nella seconda sintassi, l'argomento *funk()* consente a un programma di memorizzare la stringa ricevuta come definizione di una funzione. Questa sintassi opera come se il programma avesse eseguito il comando:

Esempio: richiedere il valore attuale del sensore di livello luminosità integrato dell'hub. Utilizzare **Get** per recuperare il valore e assegnarlo alla variabile *lightval*.

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	0.347922

Incorporare la richiesta LEGGI all'interno del comando **Get**.

Get "READ BRIGHTNESS", <i>lightval</i>	Done
<i>lightval</i>	0.378441

Definire $\text{funz}(\text{arg1}, \dots, \text{argn}) = \text{stringa ricevuta}$

Il programma può quindi usare la funzione definita $\text{funz}()$.

Nota: è possibile utilizzare il comando **Get** all'interno di un programma definito dall'utente, ma non di una funzione.

Nota: vedere anche **GetStr**, pagina 71 e **Send**, pagina 146.

getDenom() (Ottieni/restituisci denominatore)

Catalogo >

getDenom(Frazione1) ⇒ valore

Trasforma l'argomento in un'espressione con denominatore comune ridotto e successivamente restituisce il denominatore.

$x:=5; y:=6$	6
$\text{getDenom}\left(\frac{x+2}{y-3}\right)$	3
$\text{getDenom}\left(\frac{2}{7}\right)$	7
$\text{getDenom}\left(\frac{1}{x} + \frac{y^2+y}{y^2}\right)$	30

getKey()

Catalogo >

getKey([0|1]) ⇒ returnString

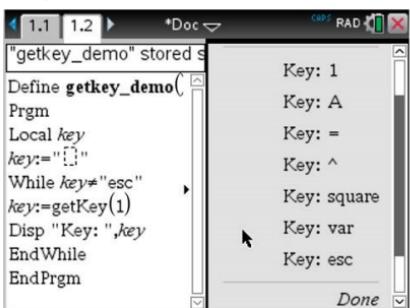
Descrizione: **getKey()** - consente al programma TI-Basic di ottenere l'input della tastiera - palmare, desktop ed emulatore su desktop.

Esempio:

- `keypressed := getKey()` restituirà un codice o una stringa vuota, se non è stato premuto alcun tasto. Questa chiamata verrà restituita immediatamente.
- `keypressed := getKey(1)` attenderà finché non verrà premuto un tasto. Questa chiamata metterà in pausa l'esecuzione del programma finché non verrà premuto un tasto.

getKey()

Esempio:



Gestione delle battute di tasto:

Tasto palmare/emulatore	Desktop	Restituire un valore
Esc	Esc	"esc"
Touchpad - Clic in alto	n/d	"up"
Acceso	n/d	"home"
Scratchapp	n/d	"scratchpad"
Touchpad - Clic a sinistra	n/d	"left"
Touchpad - Clic al centro	n/d	"center"
Touchpad - Clic a destra	n/d	"right"
Doc	n/d	"doc"
Tabulazione	Tabulazione	"tab"
Touchpad - Clic in basso	Freccia giù	"down"
Menu	n/d	"menu"
Ctrl	Ctrl	nessun valore restituito
Maiusc	Maiusc	nessun valore restituito
Var	n/d	"var"
Canc	n/d	"del"
=	=	"="
trig	n/d	"trig"
da 0 a 9	0-9	"0" ... "9"
Modelli	n/d	"template"
Catalogo	n/d	"cat"
^	^	"^"
X^2	n/d	"square"
/ (tasto per la divisione)	/	"/"
* (tasto per la moltiplicazione)	*	"*"

Tasto palmare/emulatore	Desktop	Restituire un valore
e^x	n/d	"exp"
10^x	n/d	"10power"
+	+	"+"
-	-	"_"
(("("
))	")"
.	.	".."
(-)	n/d	"-" (segno negativo)
Invio	Invio	"enter"
ee	n/d	"E" (notazione scientifica E)
a - z	a-z	alpha = lettera premuta (minuscola) ("a" - "z")
Maiusc a-z	Maiusc a-z	alpha = lettera premuta "A" - "Z"
		Nota: Ctrl-Maiusc attiva il blocco delle maiuscole
?!	n/d	"?!"
pi	n/d	"pi"
Contrassegno	n/d	nessun valore restituito
,	,	",,"
Return	n/d	"return"
spazio	spazio	" " (spazio)
Inaccessibile	Tasti con caratteri speciali @,!,&, ecc.	Il carattere viene restituito
n/d	Tasti funzione	Nessun carattere restituito
n/d	Tasti speciali di controllo desktop	Nessun carattere restituito
Inaccessibile	Non sono disponibili altri tasti desktop sulla	Lo stesso carattere restituito in Note (non in riquadro matematica)

Tasto palmare/emulatore	Desktop calcolatrice, mentre getKey() attende una battuta. ({, }, ;, ...)	Restituire un valore
-------------------------	--	----------------------

Nota: è importante notare che la presenza di **getKey()** in un programma cambia il modo in cui certi eventi vengono gestiti dal sistema. Alcuni di questi sono descritti qui di seguito.

Terminare il programma e l'evento Handle - Proprio come se l'utente dovesse uscire dal programma premendo il tasto **ON**

"**Support**" qui sotto significa - il sistema funziona come previsto - l'esecuzione del programma continua.

Evento	Dispositivo	Desktop - Software TI-Nspire™ per studenti
Test rapido	Terminare il programma, l'evento handle	Come per il palmare (solo TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)
Gestione remota dei file (incl. inviare il file 'Exit Press 2 Test' da un altro palmare o desktop-palmare)	Terminare il programma, l'evento handle	Come per il palmare. (solo TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)
Fine classe	Terminare il programma, l'evento handle	Servizio di assistenza (solo TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)

Evento	Dispositivo	Desktop - Tutte le versioni di TI-Nspire™
TI-Innovator™ Hub connettere/disconnettere	Supporto - Può inviare comandi a TI-Innovator™ Hub. Dopo la disconnessione dal programma, TI-Innovator™ Hub continua a funzionare con il palmare.	Come per il palmare

getLangInfo() (Ottieni informazioni sulla lingua)

Catalog >

getLangInfo()⇒*stringa*

getLangInfo()

"en"

Restituisce una stringa che corrisponde all'abbreviazione della lingua attiva corrente. Può essere utilizzato, ad esempio, in un programma o in una funzione per determinare la lingua corrente.

Inglese = "en"

Danese = "da"

Tedesco = "de"

Finlandese = "fi"

Francese = "fr"

Italiano = "it"

Olandese = "nl"

Olandese - Belgio = "nl_BE"

Norvegese = "no"

Portoghese = "pt"

Spagnolo = "es"

Svedese = "sv"

getLockInfo()

Catalogo >

getLockInfo(*Var*)⇒*valore*

Restituisce lo stato bloccato/sbloccato corrente della variabile *Var*.

valore =0: *Var* è sbloccata o non esiste.

valore =1: *Var* è bloccata e non può essere modificata o eliminata.

Vedere **Lock**, pagina 93 e **unLock**, pagina 179.

<i>a:=65</i>	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a:=75</i>	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a:=75</i>	75
DelVar <i>a</i>	Done

getMode(*interoNomeModo*)⇒*valore*

getMode(0)⇒*lista*

getMode(*interoNomeModo*) restituisce un valore che rappresenta l'impostazione corrente del modo *interoNomeModo*.

getMode(0) restituisce una lista contenente coppie di numeri. Ciascuna coppia è costituita da un numero intero per il modo e da un numero intero per l'impostazione.

Per un elenco dei modi e delle relative impostazioni, vedere la tabella seguente.

Se si salvano le impostazioni con **getMode(0) → var**, è possibile utilizzare **setMode(var)** in una funzione o in un programma per ripristinare temporaneamente le impostazioni solo all'interno dell'esecuzione della funzione o del programma. Vedere **setMode()**, pagina 149.

getMode(0)	{ 1,7,2,1,3,1,4,1,5,1,6,1,7,1 }
getMode(1)	7
getMode(7)	1

Nome modo	Intero modo	Interi impostazioni
Mostra cifre (Display digits)	1	1=Mobile, 2=Mobile1, 3=Mobile2, 4=Mobile3, 5=Mobile4, 6=Mobile5, 7=Mobile6, 8=Mobile7, 9=Mobile8, 10=Mobile9, 11=Mobile10, 12=Mobile11, 13=Mobile12, 14=Fissa0, 15=Fissa1, 16=Fissa2, 17=Fissa3, 18=Fissa4, 19=Fissa5, 20=Fissa6, 21=Fissa7, 22=Fissa8, 23=Fissa9, 24=Fissa10, 25=Fissa11, 26=Fissa12
Angolo (Angle)	2	1=Radiani, 2=Gradi, 3=Gradanti
Formato esponenziale (Exponential Format)	3	1=Normale, 2=Scientifico, 3=Tecnico
Reale o Complesso (Real or Complex)	4	1=Reale, 2=Rettangolare, 3=Polare
Auto o Approssimato (Auto or Approx.)	5	1=Auto, 2=Approssimato

Nome modo	Intero modo	Interi impostazioni
Formato vettoriale (Vector Format)	6	1=Rettangolare, 2=Cilindrico, 3=Sferico
Base	7	1=Decimale, 2=Esadecimale, 3=Binario

getNum() (Ottieni/restituisce numeratore)

Catalogo > 

getNum(Frazione l)⇒valore

Trasforma l'argomento in un'espressione con denominatore comune ridotto e successivamente restituisce il numeratore.

$x:=5: y:=6$	6
$\text{getNum}\left(\frac{x+2}{y-3}\right)$	7
$\text{getNum}\left(\frac{2}{7}\right)$	2
$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$	11

GetStr

Menu Hub

GetStr[promptString,] var[, statusVar] Per degli esempi vedere **Get**.

**GetStr[promptString,] funz(arg1, ...argn)
, statusVar]**

Istruzione di programmazione: funziona allo stesso modo del comando **Get**, tranne per il fatto che il valore ricevuto viene sempre interpretato come una stringa. A differenza di questo, il comando **Get** interpreta la risposta come espressione a meno che non sia racchiusa tra virgolette ("").

Nota: vedere anche **Get**, pagina 64 e **Send**, pagina 146.

getType()

Catalogo >

getType(var)⇒stringa

Restituisce una stringa che indica il tipo di dati della variabile *var*.

Se *var* non è stata definita, restituisce la stringa "NONE" (NESSUNA).

{1,2,3}→temp	{1,2,3}
getType(temp)	"LIST"
3·i→temp	3·i
getType(temp)	"EXPR"
DelVar temp	Done
getType(temp)	"NONE"

getVarInfo() (Ottieni informazioni variabile)

Catalog >

getVarInfo()⇒matrice o stringa

getVarInfo

(StringaNomeLibreria)⇒matrice o stringa

getVarInfo() restituisce una matrice di informazioni (nome di variabile, tipo, accessibilità della libreria e stato bloccato/sbloccato) per tutte le variabili e gli oggetti libreria definiti nell'attività corrente.

Se non ci sono variabili definite, **getVarInfo()** restituisce la stringa "NONE".

getVarInfo

(StringaNomeLibreria) restituisce una matrice di informazioni per tutti gli oggetti libreria definiti nella libreria

StringaNomeLibreria.

StringaNomeLibreria deve essere una stringa (testo racchiuso tra virgolette) o una variabile stringa.

Se la libreria *StringaNomeLibreria* non esiste, si produce un errore.

getVarInfo()	"NONE"
Define x=5	Done
Lock x	Done
Define LibPriv y={1,2,3}	Done
Define LibPub z(x)=3·x ² -x	Done
getVarInfo()	$\begin{bmatrix} x & \text{"NUM"} & \{ \} & 1 \\ y & \text{"LIST"} & \text{"LibPriv"} & 0 \\ z & \text{"FUNC"} & \text{"LibPub"} & 0 \end{bmatrix}$
getVarInfo(tmp3)	"Error: Argument must be a string"
getVarInfo("tmp3")	$\begin{bmatrix} \text{volcyL2} & \text{"NONE"} & \text{"LibPub"} & 0 \end{bmatrix}$

getVarInfo() (Ottieni informazioni variabile)

Catalog >

Notare l'esempio sulla sinistra, in cui il risultato di **getVarInfo()** è assegnato alla variabile *vs*. Se si tenta di visualizzare la riga 2 o la riga 3 di *vs* viene restituito un errore "Invalid list or matrix (Lista o matrice non valida)" perché almeno uno degli elementi di queste righe (ad esempio, variabile *b*) viene ricalcolato in una matrice.

Questo errore potrebbe ripresentarsi quando si utilizza *Ans* per ricalcolare un risultato di **getVarInfo()**.

Questo errore viene generato perché la versione corrente del software non supporta una struttura di matrice generalizzata quando un elemento di una matrice può essere o una matrice o una lista.

<i>a:=1</i>	1
<i>b:=[1 2]</i>	[1 2]
<i>c:=[1 3 7]</i>	[1 3 7]
<i>vs:=getVarInfo()</i>	$\begin{bmatrix} a & \text{"NUM"} & "[1 2]" & 0 \\ b & \text{"MAT"} & "[1 2]" & 0 \\ c & \text{"MAT"} & "[1 3 7]" & 0 \end{bmatrix}$
<i>vs[1]</i>	[1 "NUM" "[1 2]" 0]
<i>vs[1,1]</i>	1
<i>vs[2]</i>	"Error: Invalid list or matrix"
<i>vs[2,1]</i>	[1 2]

Goto (Vai a)

Catalogo >

Goto *nomeEtichetta*

Trasferisce il controllo all'etichetta *nomeEtichetta*.

nomeEtichetta deve essere definito nella stessa funzione mediante un'istruzione **Lbl**.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define *g()*=Func Done
Local *temp,i*
 $0 \rightarrow \text{temp}$
 $1 \rightarrow i$
Lbl *top*
 $\text{temp}+i \rightarrow \text{temp}$
If $i < 10$ Then
 $i+1 \rightarrow i$
Goto *top*
EndIf
Return *temp*
EndFunc

g() 55

►Grad (Gradi o Gradi centesimali)

Catalogo >

Espr1 ► **Grad**=*espressione*

In modalità angolo in gradi:

Converte *Espr1* in una misura di angolo in gradi.

(1.5)►Grad (1.66667)^g

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>**Grad**.

In modalità angolo in radianti:

I

identity()

Catalogo >

identity(*Intero*) \Rightarrow matriceRestituisce la matrice identità con la dimensione di *Intero*.*Intero* deve essere un numero intero positivo.

identity(4)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
If

Catalogo >

If *EsprBooleana*
 Istruzione

Define $g(x)=\text{Func}$ *Done*
 If $x < 0$ Then
 Return x^2
 EndIf
 EndFunc

If *EsprBooleana* **Then**
 Blocco
EndIfSe il valore di *EsprBooleana* è vero, viene eseguita la singola istruzione *Istruzione* o il blocco di istruzioni *Blocco* prima di procedere con l'esecuzione.Se il valore di *EsprBooleana* è falso, la funzione continua senza eseguire l'istruzione o il blocco di istruzioni.*Blocco* può essere una singola espressione o una serie di espressioni separate dal carattere ":".**Nota per l'inserimento dell'esempio:** per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto. $g(-2)$

4

If**If** *EsprBooleana* **Then** *Blocco1***Else** *Blocco2***EndIf**

Se il valore di *EsprBooleana* è vero, viene eseguito *Blocco1* e successivamente viene saltato *Blocco2*.

Se il valore di *EsprBooleana* è falso, viene saltato *Blocco1* ma viene eseguito *Blocco2*.

Blocco1 e *Blocco2* possono essere una singola istruzione.

If *EsprBooleana1* **Then** *Blocco1***If** *EsprBooleana2* **Then** *Blocco2*

:

ElseIf *EsprBooleanaN* **Then** *BloccoN***EndIf**

Permette il passaggio a un'altra funzione. Se il valore di *EsprBooleana1* è vero, viene eseguito *Blocco1*. Se *EsprBooleana1* è falso, calcola *EsprBooleana2* e così via.

Define $g(x) = \text{Func}$ *Done*If $x < 0$ ThenReturn $\neg x$

Else

Return x

EndIf

EndFunc

 $g(12)$

12

 $g(-12)$

12

ifFn()

Catalogo >

ifFn(*EsprBooleana*, *Valore_se_vero* [, *Valore_se_falso* [, *Valore_se_sconosciuto*]]) \Rightarrow espressione, lista o matrice

Calcola l'espressione booleana

EsprBooleana (o ciascun elemento di *EsprBooleana*) e produce un risultato sulla base delle seguenti regole:

- *EsprBooleana* può verificare un singolo valore, una lista o una matrice.
- Se un elemento di *EsprBooleana* dà come risultato una condizione vera,

ifFn($\{1, 2, 3\} < 2.5, \{5, 6, 7\}, \{8, 9, 10\}$)

{5, 6, 10}

Il valore di verifica di **1** è minore di 2.5, così l'elemento

Valore_se_vero corrispondente di **5** viene copiato nella lista del risultato.

Il valore di verifica di **2** è minore di 2.5, così l'elemento

viene restituito l'elemento corrispondente di *Valore se vero*.

- Se un elemento di *EsprBooleana* dà come risultato una condizione falsa, viene restituito l'elemento corrispondente di *Valore_se_falso*. Se *Valore_se_falso* è omesso, viene restituito *undef*.
 - Se un elemento di *EsprBooleana* non dà come risultato una condizione vera né una condizione falsa, viene restituito l'elemento corrispondente di *Valore_se_sconosciuto*. Se *Valore_se_sconosciuto* è omesso, viene restituito *undef*.
 - Se il secondo, terzo o quarto argomento della funzione **ifFn()** è una singola espressione, la verifica booleana viene applicata a tutte le posizioni in *EsprBooleana*.

Nota: se l'istruzione *EsprBooleana* semplificata implica una lista o una matrice, tutti gli altri argomenti della lista o della matrice devono avere uguali dimensioni e il risultato avrà uguali dimensioni.

imag()

imag(Valore1) ⇒ valore

Restituisce la parte immaginaria dell'argomento.

imag(Lista) \Rightarrow lista

Restituisce una lista delle parti non reali degli elementi.

imag(*Matrice1***)** ⇒ *matrice*

Restituisce una matrice delle parti immaginarie degli elementi.

Valore_se_vero corrispondente di 6 viene copiato nella lista del risultato.

Il valore di verifica di **3** è minore di 2.5, così l'elemento *Valore_se_falso* corrispondente di **10** viene copiato nella lista del risultato.

ifFn($\{1,2,3\} < 2.5, 4, \{8,9,10\}\}$) $\{4,4,10\}$

Valore_se_vero è un singolo valore e corrisponde a qualsiasi posizione selezionata.

ifFn({1,2,3}<2.5,{5,6,7}) {5,6,undefined}

Valore_se_falso non è specificato. Viene utilizzato Undef.

```
ifFn({2,"a"}<2.5,{6,7},{9,10},"err")  
{6,"err"}
```

Un elemento selezionato da *Valore_se_vero*. Un elemento selezionato da *Valore_se_sconosciuto*.

Catalogo >

imag(1+2·i)

2

imag{ $\{-3, 4 - i, i\}$ }

$$\{0,-1,1\}$$

imag{[1 2]}

$$\begin{bmatrix} 0 & 0 \end{bmatrix}$$

inString()**Catalogo >**

inString(stringaRicerca, sottoStringa[, Inizio]) \Rightarrow intero

Restituisce la posizione del carattere nella stringa *stringaRicerca* dal quale inizia per la prima volta la stringa *sottoStringa*.

Inizio, se incluso, specifica la posizione del carattere all'interno di *stringaRicerca* in cui comincia la stringa. L'impostazione predefinita è 1 (il primo carattere di *stringaRicerca*).

Se *stringaRicerca* non contiene *sottoStringa* o *Inizio* è $>$ della lunghezza di *stringaRicerca*, viene restituito zero.

inString("Hello there","the")	7
inString("ABCEFG","D")	0

int()**Catalogo >**

int(Valore) \Rightarrow intero
int(Lista) \Rightarrow lista
int(MatriceI) \Rightarrow matrice

int(-2.5)	-3.
int([-1.234 0 0.37])	[-2. 0 0.]

Restituisce il più grande dei numeri interi che è minore o uguale all'argomento.
Questa funzione è identica a **floor()**.

L'argomento può essere un numero reale o complesso.

Con una lista o con una matrice, restituisce il numero intero maggiore di ciascuno degli elementi.

intDiv()**Catalogo >**

intDiv(Numer01, Numer02) \Rightarrow intero
intDiv(Lista1, Lista2) \Rightarrow lista
intDiv(Matrice1, Matrice2) \Rightarrow matrice

intDiv(-7,2)	-3
intDiv(4,5)	0
intDiv({12,-14,-16},{5,4,-3})	{2,-3,5}

Restituisce la parte intera con segno di $(Numer01 \div Numer02)$.

intDiv()

Catalogo >

Con liste e matrici, restituisce la parte intera con segno di (argomento1 ÷ argomento2) per ciascuna coppia di elementi.

interpolare()

Catalogo >

interpolare(valoreX, listaX, listaY, listaYprime) ⇒ lista

Questa funzione consente di fare quanto segue:

Dati $listaX$, $lista Y=f(listaX)$, e $listaYprime=f'(listaX)$ per una funzione f non nota, viene utilizzata una interpolante cubica per approssimare la funzione f al $valoreX$. Si assume che $listaX$ è una lista di numeri monotonicamente crescenti o decrescenti, ma questa funzione può restituire un valore anche quando non c'è. Questa funzione scorre $listaX$ cercando un intervallo $[listaX[i], listaX[i+1]]$ che contenga $valoreX$. Se trova un intervallo di questo tipo, restituisce un valore interpolato per $f(valoreX)$; altrimenti, restituisce **undef**.

$listaX$, $listaY$ e $listaPrimiY$ devono avere la stessa dimensione ≥ 2 e contenere espressioni che vengono semplificate in numeri.

$valoreX$ può essere un numero o una lista di numeri.

Equazione differenziale:

$$y' = -3 \cdot y + 6 \cdot t + 5 \text{ e } y(0) = 5$$

$$\begin{aligned} rk := & \text{rk23}\{-3 \cdot y + 6 \cdot t + 5, y, \{0, 10\}, 5, 1\} \\ & \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 5. & 3.19499 & 5.00394 & 6.99957 & 9.00593 \end{bmatrix} \end{aligned}$$

Per vedere l'intero risultato, premere \blacktriangleleft , quindi utilizzare \blacktriangleleft e \blacktriangleright per spostare il cursore.

Utilizzare la funzione **interpolate()** per calcolare i valori della funzione per $listavalorix$:

$$\begin{aligned} xvalueList := & \text{seq}\{i, i, 0, 10, 0.5\} \\ & \{0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10\} \\ xList := & \text{mat}\mapsto\text{list}(rk[1]) \\ & \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\ yList := & \text{mat}\mapsto\text{list}(rk[2]) \\ & \{5, 3.19499, 5.00394, 6.99957, 9.00593, 10.997\} \\ yPrimeList := & -3 \cdot y + 6 \cdot t + 5 | y=yList \text{ and } t=xList \\ & \{-10, -1.41503, 1.98819, 2.00129, 1.98221, 2.006\} \\ \text{interpolate}\{ & xvalueList, xList, yList, yPrimeList\} \\ & \{5, 2.67062, 3.19499, 4.02782, 5.00394, 6.0001\} \end{aligned}$$

invχ²()

Catalogo >

invχ²(Area, df)

invChi2(Area, gl)

Calcola la funzione della probabilità χ^2 (chi quadrato) cumulativa inversa specificata dal grado di libertà, df , per una data $Area$ sotto la curva.

invF(*Area,glNumer,glDenom*)**invF(*Area,glNumer,glDenom*)**

Calcola la funzione della distribuzione F cumulativa inversa specificata da *glNumer* e *glDenom* per una data *Area* sotto la curva.

invBinom()**invBinom****(*CumulativeProb, numProve, Prob, OutputForm*)**⇒ scalare o matrice

Dato il numero di tentativi (*NumTrials*) e la probabilità di successo di ogni tentativo (*Prob*), questa funzione restituisce il numero minimo di successi *k*, in modo che la probabilità cumulativa di successi *k* sia maggiore o uguale alla probabilità cumulativa (*CumulativeProb*) data.

OutputForm=0 visualizza il risultato come un valore scalare (impostazione predefinita).

OutputForm=1, visualizza il risultato come una matrice.

Esempio: Mary e Kevin stanno giocando a dadi. Mary deve indovinare il numero massimo di volte che il numero 6 può apparire in 30 lanci. Mary vince se il numero 6 appare il numero di volte indicate. Inoltre, più piccolo è il numero indovinato, maggiori saranno le sue vincite. Qual è il più piccolo numero che Mary può indovinare se desidera che le probabilità di vincere siano maggiori del 77%?

$$\begin{aligned} \text{invBinom}\left(0.77, 30, \frac{1}{6}\right) &= 6 \\ \text{invBinom}\left(0.77, 30, \frac{1}{6}, 1\right) &= \begin{bmatrix} 5 & 0.616447 \\ 6 & 0.776537 \end{bmatrix} \end{aligned}$$

invBinomN()**invBinomN(*CumulativeProb, Prob, NumSuccess, OutputForm*)**⇒ scalare o matrice

Data la probabilità di successo di ogni tentativo (*Prob*) e il numero di successi (*NumSuccess*), questa funzione restituisce il numero minimo di tentativi *N*, in modo che la probabilità cumulativa di successi *x* sia minore o uguale alla probabilità cumulativa (*CumulativeProb*) data.

OutputForm=0 visualizza il risultato come un valore scalare (impostazione predefinita).

OutputForm=1, visualizza il risultato come una matrice.

Esempio: Monique si sta allenando a tirare in porta a netball. Monique sa per esperienza di avere il 70% di possibilità di realizzare ogni tiro e decide di continuare ad allenarsi fino a aver segnato 50 goal. Quanti tiri deve tentare per essere sicura che le probabilità di segnare almeno 50 goal siano maggiori di 0,99?

$$\begin{aligned} \text{invBinomN}(0.01, 0.7, 49) &= 86 \\ \text{invBinomN}(0.01, 0.7, 49, 1) &= \begin{bmatrix} 85 & 0.010451 \\ 86 & 0.00709 \end{bmatrix} \end{aligned}$$

invNorm() (Distribuzione normale cumulativa inversa)

Catalogo >

invNorm(*Area*[, μ [, σ]])

Calcola la funzione della distribuzione normale cumulativa inversa per una data *Area* sotto la curva della distribuzione normale specificata da μ e σ .

invt() (Funzione della probabilità t di Student)

Catalogo >

invt(*Area*,*gl*)

Calcola la funzione della probabilità cumulativa inversa t di Student specificata dal grado di libertà, *gl* per una data *Area* sotto la curva.

iPart()

Catalogo >

iPart(*Numero*) \Rightarrow intero

iPart(*ListaL*) \Rightarrow lista

iPart(*MatriceL*) \Rightarrow matrice

Restituisce la parte intera dell'argomento.

Per una lista o per una matrice, restituisce la parte intera di ciascun argomento.

L'argomento può essere un numero reale o complesso.

iPart(-1.234)	-1.
iPart($\left\{ \frac{3}{2}, -2.3, 7.003 \right\}$)	{1,-2,7.}

irr()

Catalogo >

irr(*CF0*,*CFLista* [,*CFFreq*]) \Rightarrow valore

Funzione finanziaria che calcola l'indice di rendimento interno di un investimento (Internal Rate of Return).

CF0 è il cash flow iniziale al tempo 0; deve essere un numero reale.

CFLista è una lista di importi di cash flow dopo il cash flow iniziale *CF0*.

<i>list1</i> := {6000,-8000,2000,-3000}	{6000, 8000,2000, -3000}
<i>list2</i> := {2,2,2,1}	{2,2,2,1}
irr(5000, <i>list1</i> , <i>list2</i>)	-4.64484

CFFreq è una lista opzionale in cui ciascun elemento specifica la frequenza di occorrenza di un importo di cash flow raggruppato (consecutivo), che è l'elemento corrispondente di *CFLista*. L'impostazione predefinita è 1; è possibile inserire valori che siano numeri interi positivi < 10000.

Nota: Vedere anche **mirr()**, pagina 102.

isPrime()

isPrime(*Numero*) ⇒ espressione costante booleana

Restituisce vero o falso per indicare se *numero* è un numero intero ≥ 2 divisibile solo per se stesso e per 1.

Se *Numero* ha più di 306 cifre e non ha fattori ≤ 1021 , **isPrime(*Numero*)** visualizza un messaggio di errore.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

isPrime(5)	true
isPrime(6)	false

Funzione per trovare il numero primo successivo dopo il numero specificato.

Define nextprim(<i>n</i>)=Func	Done
Loop	
<i>n</i> +1 → <i>n</i>	
If isPrime(<i>n</i>)	
Return <i>n</i>	
EndLoop	
EndFunc	
nextprim(7)	11

isVoid()

isVoid(*Var*) ⇒ espressione costante booleana

isVoid(*Espr*) ⇒ espressione costante booleana

isVoid(*Lista*) ⇒ lista di espressione costante booleana

Restituisce vero o falso per indicare se l'argomento è un tipo di dati vuoto.

Per ulteriori informazioni sugli elementi vuoti, vedere pagina 233.

<i>a</i> :=_	_
isVoid(<i>a</i>)	true
isVoid({1,_,3})	{ false,true,false }

Lbl (Etichetta)**Lbl** *nomeEtichetta*

Definisce un'etichetta chiamata *nomeEtichetta* in una funzione.

È possibile utilizzare un'istruzione **Goto** *nomeEtichetta* per trasferire il controllo del programma all'istruzione immediatamente successiva all'etichetta.

nomeEtichetta deve soddisfare gli stessi requisiti validi per i nomi delle variabili.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Catalogo > Define *g()*=Func*Done*Local *temp,i*0 → *temp*1 → *i*Lbl *top**temp*+*i* → *temp*If *i*<10 Then*i*+1 → *i*Goto *top*

EndIf

Return *temp*

EndFunc

g()

55

Icm() (Minimo comune multiplo)**Catalogo >** **Icm**(*Numero1, Numero2*)⇒*espressione***Icm**(*List1, Lista2*)⇒*lista***Icm**(*Matrice1, Matrice2*)⇒*matrice*

Restituisce il minimo comune multiplo (lcm) di due argomenti. Il lcm di due frazioni è il **minimo comune multiplo** dei loro numeratori diviso per il **massimo comune divisore** (gcd) dei loro denominatori. Il lcm dei numeri frazionari a virgola mobile è il loro prodotto.

Per due liste o matrici, restituisce i minimi comuni multipli dei corrispondenti elementi.

lcm(6,9)

18

lcm{ $\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}\right\}$ { $\frac{2}{3}, 14, 80$ }**left() (Sinistra)****Catalogo >** **left**(*stringaOrigine*[, *Num*])⇒*stringa*

left("Hello",2)

"He"

Restituisce i caratteri *Num* più a sinistra contenuti nella stringa di caratteri *stringaOrigine*.

left() (Sinistra)

Catalogo >

Se si omette *Num*, restituisce l'intera *stringaOrigine*.

left(Lista1[, Num])⇒*lista*

`left({1,3,-2,4},3)`

{1,3,-2}

Restituisce gli elementi *Num* più a sinistra contenuti in *Lista1*.

Se si omette *Num*, restituisce l'intera *Lista1*.

left(Confronto)⇒espressione

Restituisce il primo membro di un'equazione o di una disequazione.

libShortcut() (Collegamento a libreria)

Catalogo >

**libShortcut(stringaNomeLibr,
stringaNomeCollegamento[,
LibPrivFlag])**⇒*lista di variabili*

Crea una gruppo di variabili nell'attività corrente che contiene i riferimenti a tutti gli oggetti nel documento libreria specificato *stringaNomeLibr*. Aggiunge inoltre i membri del gruppo al menu Variables (Variabili). È quindi possibile fare riferimento a ciascun oggetto utilizzando la relativa *stringaNomeCollegamento*.

Impostare *LibPrivFlag=0* per escludere oggetti libreria privata (default)

Impostare *LibPrivFlag=1* per includere oggetti libreria privata

Per copiare un gruppo di variabili, vedere **CopyVar**, pagina 26.

Per eliminare un gruppo di variabili, vedere **DelVar**, pagina 41.

Questo esempio presuppone un documento libreria memorizzato e aggiornato, denominato **linalg2**, che contiene oggetti definiti come *clearmat*, *gauss1* e *gauss2*.

`getVarInfo("linalg2")`

`clearmat "FUNC" "LibPub"
gauss1 "PRGM" "LibPriv"
gauss2 "FUNC" "LibPub"`

`libShortcut("linalg2","la")`

{la.clearmat,la.gauss2}

`libShortcut("linalg2","la",1)`

{la.clearmat,la.gauss1,la.gauss2}

LinRegBx (Regressione lineare)

Catalogo >

LinRegBx X,Y[,Freq[,Categoria,Includi]]

Calcola la regressione lineare $y = a + b \cdot x$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a+b \cdot x$
stat.a, stat.b	Coefficienti di regressione
stat.r ²	coefficiente di determinazione
stat.r	Coefficiente di correlazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

LinRegMx X,Y[,Freq[,Categoria,Includi]]

Calcola la regressione lineare $y = m \cdot x + b$ sulle liste X e Y con frequenza $Freq$. Il riepilogo dei risultati è memorizzato nella variabile `stat.results`. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e Y sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di X e Y . Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti X e Y .

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $m \cdot x + b$
stat.m, stat.b	Coefficienti di regressione
stat.r ²	coefficiente di determinazione
stat.r	Coefficiente di correlazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

LinRegtIntervals $X, Y[, F[, 0 [, livelloConfidenza]]]$

Per pendenza Calcola un intervallo di confidenza di livello C per la pendenza.

LinRegtIntervals $X, Y[, F[, 1, ValX [, livelloConfidenza]]]$

Per risposta. Calcola un valore y previsto, un intervallo di previsione del livello C per una singola osservazione e un intervallo di confidenza del livello C per la risposta media.

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere la stessa dimensione.

X e *Y* sono liste di variabili indipendenti e dipendenti.

F è una lista opzionale di valori di frequenza. Ciascun elemento di *F* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a+b \cdot x$
stat.a, stat.b	Coefficienti di regressione
stat.df	Gradi di libertà
stat.r ²	Coefficiente di determinazione
stat.r	Coefficiente di correlazione
stat.Resid	Residui della regressione

Solo tipo per pendenza

Variabile di output	Descrizione
[stat.CLower, stat.CUpper]	Intervallo di confidenza per la pendenza
stat.ME	Margine di errore dell'intervallo di confidenza
stat.SESlope	Errore standard della pendenza
stat.s	Errore standard sulla linea

Solo tipo per risposta

Variabile di output	Descrizione
[stat.CLower, stat.CUpper]	Intervallo di confidenza per la risposta media
stat.ME	Margine di errore dell'intervallo di confidenza
stat.SE	Errore standard della risposta media
[stat.LowerPred, stat.UpperPred]	Intervallo di previsione per una singola osservazione
stat.MEPred	Margine di errore dell'intervallo di previsione
stat.SEPred	Errore standard per la previsione
stat. \hat{y}	$a + b \cdot \text{ValX}$

LinRegtTest (t Test regressione lineare)

Catalogo > 

LinRegtTest $X, Y[, Freq[, Ipotesi]]$

Calcola una regressione lineare sulle liste X e Y e un t test sul valore della pendenza β e il coefficiente di correlazione ρ per l'equazione $y = \alpha + \beta x$. Viene verificata l'ipotesi nulla $H_0: \beta = 0$ (in modo equivalente, $\rho = 0$) in relazione a una di tre ipotesi alternative.

Tutte le liste devono avere la stessa dimensione.

X e Y sono liste di variabili indipendenti e dipendenti.

$Freq$ è una lista opzionale di valori di frequenza. Ciascun elemento di $Freq$ specifica la frequenza di occorrenza di ogni dato corrispondente di X e Y . Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Ipotesi è un valore opzionale che specifica una di tre ipotesi alternative rispetto alla quale verrà testata l'ipotesi nulla ($H_0: \beta = \rho = 0$).

Per $H_a: \beta \neq 0$ e $\rho \neq 0$ (default), impostare *Ipotesi*=0

Per $H_a: \beta < 0$ e $\rho < 0$, impostare *Ipotesi*<0

Per $H_a: \beta > 0$ e $\rho > 0$, impostare *Ipotesi*>0

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a + b \cdot x$
stat.t	Statistica T per il test di significatività
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Gradi di libertà
stat.a, stat.b	Coefficienti di regressione
stat.s	Errore standard sulla linea
stat.SESlope	Errore standard della pendenza
stat.r ²	Coefficiente di determinazione
stat.r	Coefficiente di correlazione
stat.Resid	Residui della regressione

linSolve()

Catalogo >

linSolve(*SistemaDiEqLineari, Var1, Var2, ...)***=>***lista*

linSolve(*EqLineare1 and EqLineare2 and ..., Var1, Var2, ...)***=>***lista*

linSolve({*EqLineare1, EqLineare2, ...*}, *Var1, Var2, ...*)**=>***lista*

linSolve(*SistemaDiEqLineari, {Var1, Var2, ...}*)**=>***lista*

linSolve(*EqLineare1 and EqLineare2 and ..., {Var1, Var2, ...}*)**=>***lista*

linSolve({*EqLineare1, EqLineare2, ...*}, *{Var1, Var2, ...}*)**=>***lista*

Restituisce una lista di soluzioni per le variabili *Var1, Var2, ...*

Il primo argomento deve calcolare un sistema di equazioni lineari o una singola equazione lineare. Diversamente, si ottiene un argomento errato.

Ad esempio, calcolando *linSolve(x=1 e x=2,x)* si ottiene un risultato “Argomento errato”.

$$\text{linSolve}\left(\begin{cases} 2x+4y=3 \\ 5x-3y=7 \end{cases}, \{x,y\}\right) \quad \left\{ \frac{37}{26}, \frac{1}{26} \right\}$$

$$\text{linSolve}\left(\begin{cases} 2x=3 \\ 5x-3y=7 \end{cases}, \{x,y\}\right) \quad \left\{ \frac{3}{2}, \frac{1}{6} \right\}$$

$$\text{linSolve}\left(\begin{cases} apple+4\cdot pear=23 \\ 5\cdot apple-pear=17 \end{cases}, \{apple,pear\}\right) \quad \left\{ \frac{13}{3}, \frac{14}{3} \right\}$$

$$\text{linSolve}\left(\begin{cases} apple+4\cdot \frac{pear}{3}=14 \\ -apple+pear=6 \end{cases}, \{apple,pear\}\right) \quad \left\{ \frac{36}{13}, \frac{114}{13} \right\}$$

Δlist() (Differenza in una lista)

Catalogo >

Δlist(*Listal*)**=>***lista*

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **deltaList(...)**.

Restituisce una lista contenente le differenze tra elementi consecutivi in *Listal*. Ogni elemento di *Listal* viene sottratto dal successivo elemento di *Listal*. La lista risultante è sempre composta da un elemento in meno della *Listal* originale.

$$\Delta\text{List}\{\{20, 30, 45, 70\}\} \quad \{10, 15, 25\}$$

list►mat() (Da lista a matrice)**Catalogo >** **list►mat(Lista [, elementiPerRiga])⇒matrice**Restituisce una matrice completata riga per riga con gli elementi di *Lista*.*elementiPerRiga*, se incluso, specifica il numero di elementi per riga.L'impostazione predefinita corrisponde al numero di elementi di *Lista* (una riga).Se *Lista* non completa la matrice risultante, viene aggiunta una serie di zeri.**Nota:** è possibile inserire questa funzione dalla tastiera del computer digitando**list@>mat(...).**

list►mat({1,2,3})	[1 2 3]
list►mat({1,2,3,4,5},2)	[1 2 3 4 5 0]

ln() (Logaritmo naturale)**Tasti** **ln(Valore l)⇒valore**

ln(2.)	0.693147
--------	----------

ln(Lista l)⇒lista

Restituisce il logaritmo naturale dell'argomento.

In una lista, restituisce i logaritmi naturali degli elementi.

Se la modalità del formato complesso è Reale:

ln({{-3,1,2,5}})	"Error: Non-real calculation"
------------------	-------------------------------

ln(matriceQuadrata1)⇒matriceQuadrataRestituisce il logaritmo naturale della matrice *matriceQuadrata1*. Ciò non equivale a calcolare il logaritmo naturale di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.*matriceQuadrata1* deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

Se la modalità del formato complesso è Rettangolare:

ln({{-3,1,2,5}})	{1.09861+3.14159·i, 0.182322, 1.60944}
------------------	--

In modalità angolo in radianti e in modalità formato rettangolare complesso:

ln	$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$
	1.83145+1.73485·i 0.009193-1.49086
	0.448761-0.725533·i 1.06491+0.623491·i
	-0.266891-2.08316·i 1.12436+1.79018·i

Per vedere l'intero risultato, premere **▲**,
quindi utilizzare **◀** e **▶** per spostare il
cursore.

LnReg (Regressione logaritmica)**Catalogo >** **LnReg X, Y[, [Freq] [, Categoria, Includi]]**

Calcola la regressione logaritmica $y = a + b \cdot \ln(x)$ sulle liste X e Y con frequenza $Freq$. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e Y sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di X e Y . Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti X e Y .

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a + b \cdot \ln(x)$
stat.a, stat.b	Coefficienti di regressione
stat.r ²	Coefficiente di determinazione lineare di dati trasformati
stat.r	Coefficiente di correlazione per dati trasformati ($\ln(x)$, y)

Variabile di output	Descrizione
stat.Resid	Residui associati al modello logaritmico
stat.ResidTrans	Residui associati all'adattamento lineare dei dati trasformati
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

Local (Variabile locale)

Catalogo > 

Local *Var1[, Var2] [, Var3] ...*

Definisce le *var* specificate come variabili locali. Tali variabili esistono solamente durante il calcolo di una funzione e vengono cancellate al termine dell'esecuzione di tale funzione.

Nota: le variabili locali permettono di risparmiare memoria in quanto esistono solo temporaneamente. Inoltre, esse non influiscono sui valori delle variabili globali esistenti. Le variabili locali devono essere utilizzate per i cicli **For** e per salvare in maniera provvisoria i valori in una funzione su diverse righe, poiché le modifiche sulle variabili globali non sono ammesse in una funzione.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

```
Define rollcount()=Func
  Local i
  1→i
  Loop
  If randInt(1,6)=randInt(1,6)
  Goto end
  i+1→i
EndLoop
Lbl end
Return i
EndFunc
Done
```

rollcount()	16
rollcount()	3

Lock

Catalogo > 

LockVar1 [, Var2] [, Var3] ...

LockVar.

Blocca le variabili o il gruppo di variabili specificate. Le variabili bloccate non possono essere modificate o eliminate.

Non è possibile bloccare o sbloccare la variabile di sistema *Ans*, inoltre non è possibile bloccare i gruppi di variabili di sistema *stat.* o *tvm*.

Nota: Il comando **Blocca (Lock)** cancella la cronologia di Annulla/Ripeti quando è applicato a variabili sbloccate.

Vedere **unLock**, pagina 179 e **getLockInfo()**, pagina 69.

a:=65	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a</i> :=75	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a</i> :=75	75
DelVar <i>a</i>	Done

log() (Logaritmo)

Tasti  

log(Valore1[,Valore2])=>*valore*

$\log_{10}(2.)$ 0.30103

log(Lista1[,Valore2])=>*lista*

$\log_4(2.)$ 0.5

Restituisce il logaritmo in base-*Valore2* dell'argomento.

$\log_3(10)-\log_3(5)$ 0.63093

Nota: vedere anche **Modello di logaritmo**, pagina 2.

Se la modalità del formato complesso è Reale:

In una lista, restituisce il logaritmo in base-*Valore2* degli elementi.

$\log_{10}(\{-3,1,2,5\})$

Se *Espr2* viene omesso, come base viene utilizzato 10.

"Error: Non-real calculation"

log(matriceQuadrata1 [,Valore2])=>*matriceQuadrata*

Se la modalità del formato complesso è Rettangolare:

$\log_{10}(\{-3,1,2,5\})$
 $\{0.477121+1.36438 \cdot i, 0.079181, 0.69897\}$

In modalità angolo in radianti e in modalità formato rettangolare complesso:

Restituisce il logaritmo in base-*Valore2* della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il logaritmo in base-*Valore2* di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

Se l'argomento base viene omesso, come base viene utilizzato 10.

$$\log_{10} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 0.795387 + 0.753438 \cdot i & 0.003993 - 0.6474 \cdot i \\ 0.194895 - 0.315095 \cdot i & 0.462485 + 0.2707 \cdot i \\ -0.115909 - 0.904706 \cdot i & 0.488304 + 0.7774 \cdot i \end{bmatrix}$$

Per vedere l'intero risultato, premere \blacktriangleleft , quindi utilizzare \blacktriangleleft e \blacktriangleright per spostare il cursore.

Logistic (Regressione logistica)

Catalogo >

Logistic *X, Y[, Freq [, Categoria, Includi]]*

Calcola la regressione logistica $y = (c/(1+a \cdot e^{-bx}))$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $c/(1+a \cdot e^{-bx})$
stat.a, stat.b, stat.c	Coefficienti di regressione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

LogisticD (Rgressione logistica)

Catalogo > 

LogisticD *X*, *Y* [, [*Iterazioni*], [*Freq*] [, *Categoria*, *Includi*]]

Calcola la regressione logistica $y = (c/(1+a \cdot e^{-bx})+d)$ sulle liste *X* e *Y* con frequenza *Freq*, utilizzando un numero specificato di *Iterazioni*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Iterazioni è un valore opzionale che specifica quante volte al massimo verrà tentata una soluzione. Se omesso, viene utilizzato 64. Di solito valori più alti danno una maggiore accuratezza ma richiedono tempi di esecuzione più lunghi, e viceversa.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $c/(1+a \cdot e^{-bx})+d)$
stat.a, stat.b, stat.c, stat.d	Coefficienti di regressione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

Loop

Loop
Blocco
EndLoop

Esegue ciclicamente le istruzioni di *Blocco*. Si noti che un ciclo viene eseguito infinite volte, se non si trovano istruzioni **Goto** o **Exit** all'interno di *Blocco*.

Blocco è una sequenza di istruzioni separate dal carattere.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

```
Define rollcount()=Func
  Local i
  1→i
  Loop
    If randInt(1,6)=randInt(1,6)
      Goto end
    i+1→i
  EndLoop
  Lbl end
  Return i
EndFunc
```

	Done
rollcount()	16
rollcount()	3

LU (Scomposizione inferiore - superiore)

Catalogo >

LU Matrice, MatriceL, MatriceU, MatriceP[,Tol]

Calcola la scomposizione LU (lower-upper, inferiore-superiore) di una matrice reale o complessa. La matrice triangolare inferiore è memorizzata in *MatriceL*, quella superiore in *MatriceU* e la matrice di permutazione (che descrive gli scambi di riga eseguiti durante i calcoli) in *MatriceP*.

$$\text{MatriceL} \cdot \text{MatriceU} = \text{MatriceP} \cdot \text{matrice}$$

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Toll*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Toll* viene ignorato.

- Se si usa **[ctrl] [enter]** oppure se si imposta la modalità **Auto** o **Approssimato** su Approssimato, i calcoli verranno eseguiti in virgola mobile.
- Se *Toll* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:
 $5\text{E}-14 \cdot \max(\dim(\text{Matrice})) \cdot \text{rowNorm}(\text{Matrice})$

L'algoritmo di scomposizione in fattori **LU** usa il pivoting parziale per lo scambio di righe.

M

mat►list() (Da matrice a lista)

Catalogo >

mat►list(Matrice)⇒lista

Restituisce una lista completata con gli elementi di *Matrice*. Gli elementi sono copiati da *Matrice* riga per riga.

$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$
LU <i>m1,lower,upper,perm</i>	<i>Done</i>
<i>lower</i>	$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
<i>upper</i>	$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
<i>perm</i>	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

mat►list([1 2 3])	{1,2,3}
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
mat►list(<i>m1</i>)	{1,2,3,4,5,6}

matlist() (Da matrice a lista)**Catalogo > **

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando `mat@>list(...)`.

max() (Massimo)**Catalogo > ****max(Valore1, Valore2)⇒espressione**

max{2.3,1.4} 2.3

max(Lista1, Lista2)⇒lista

max{{1,2},{-4,3}} {1,3}

max(Matrice1, Matrice2)⇒matrice

Restituisce il massimo di due argomenti; se questi sono due liste o matrici, restituisce una lista o matrice contenente il valore massimo di ciascuna coppia di elementi corrispondenti.

max(Lista)⇒espressione

max{{0,1,-7,1.3,0.5}} 1.3

Restituisce l'elemento massimo di *lista*.

max(Matrice1)⇒matrice

max[[1 -3 7], [-4 0 0.3]] [1 0 7]

Restituisce un vettore riga contenente l'elemento massimo di ciascuna colonna di *Matrice1*.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

Nota: vedere anche **min()**.

mean() Media**Catalogo > ****mean(Lista[, listaFreq])⇒espressione**

mean{{0.2,0.1,-0.3,0.4}} 0.26

Restituisce la media degli elementi di *Lista*.

mean{{1,2,3},{3,2,1}} 5/3

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

mean(Matrice1[, matriceFreq])⇒matrice

In modalità formato vettore rettangolare:

Restituisce un vettore riga contenente le medie di tutte le colonne di *Matrice1*.

mean() Media

Catalogo >

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Matrice1*.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

$$\text{mean} \begin{pmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{pmatrix} \quad [-0.133333 \quad 0.833333]$$

$$\text{mean} \begin{pmatrix} 1 & 0 \\ 5 & \\ -1 & 3 \\ 2 & -1 \\ 5 & 2 \end{pmatrix} \quad \begin{pmatrix} -2 & 5 \\ 15 & 6 \end{pmatrix}$$

$$\text{mean} \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \begin{pmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{pmatrix} \quad \begin{pmatrix} 47 & 11 \\ 15 & 3 \end{pmatrix}$$

median() (Mediana)

Catalogo >

median(Lista[, listaFreq]) \Rightarrow espressione

$$\text{median}\{\{0.2, 0.1, -0.3, 0.4\}\} \quad 0.2$$

Restituisce la mediana degli elementi di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

median(Matrice1[, matriceFreq]) \Rightarrow matrice

$$\text{median} \begin{pmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{pmatrix} \quad [0.4 \quad -0.3]$$

Restituisce un vettore riga contenente le mediane degli elementi delle colonne di *Matrice1*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Matrice1*.

Note:

- Tutti gli elementi nella lista o nella matrice devono essere semplificati in numeri.
- Gli elementi vuoti (nulli) di una lista o matrice vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

MedMed (Linea mediana-mediana)

Catalogo >

MedMed X,Y [, Freq] [, Categoria,

Includi]]

Calcola la linea mediana-medianay = $(m \cdot x + b)$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione linea mediana-mediana: $m \cdot x + b$
stat.m, stat.b	Coefficienti del modello
stat.Resid	Residui della linea mediana-mediana
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

mid() (In mezzo alla stringa)

Catalogo >

mid(stringaOrigine, Inizio[, Cont])⇒stringa

Restituisce *Cont* caratteri dalla stringa di caratteri *stringaOrigine*, incominciando dal numero di carattere di *Inizio*.

Se *Cont* viene omesso o se è maggiore della dimensione di *stringaOrigine*, restituisce tutti i caratteri di *stringaOrigine*, incominciando dal numero di carattere di *Inizio*.

Cont deve essere ≥ 0 . Se *Cont* = 0, viene restituita una stringa vuota.

mid(listaOrigine, Inizio [, Cont])⇒lista

Restituisce *Cont* elementi da *listaOrigine*, iniziando dal numero di elemento di *Inizio*.

Se *Cont* viene omesso o è maggiore della dimensione di *listaOrigine*, restituisce tutti gli elementi di *listaOrigine*, incominciando dal numero di elemento di *Inizio*.

Cont deve essere ≥ 0 . Se *Cont* = 0, restituisce una lista vuota.

mid(listaStringaOrigine, Inizio[, Cont])⇒lista

Restituisce *Cont* stringhe della lista delle stringhe *listaStringheOrigine*, incominciano dal numero di elemento di *Inizio*.

mid("Hello there",2)	"ello there"
mid("Hello there",7,3)	"the"
mid("Hello there",1,5)	"Hello"
mid("Hello there",1,0)	"[]"

mid({9,8,7,6},3)	{7,6}
mid({9,8,7,6},2,2)	{8,7}
mid({9,8,7,6},1,2)	{9,8}
mid({9,8,7,6},1,0)	{[]}

mid({ "A","B","C","D"},2,2)	{"B","C"}
-----------------------------	-----------

min() (Minimo)

Catalogo >

min(Valore1, Valore2)⇒espressione

min(Lista1, Lista2)⇒lista

min(Matrice1, Matrice2)⇒matrice

Restituisce il minimo di due argomenti. Se gli argomenti sono due liste o matrici, viene restituita una lista o una matrice contenente il valore minimo di ciascuna coppia di elementi corrispondenti.

min(2,3,1,4)	1.4
min({1,2},{-4,3})	{-4,2}

min() (Minimo)

Catalogo >

min(Lista)⇒espressione

Restituisce l'elemento minimo di *Lista*.

min(Matrice I)⇒matrice

Restituisce un vettore di riga contenente l'elemento minimo di ciascuna colonna di *Matrice I*.

Nota: vedere anche **max()**.

min({0,1,-7,1.3,0.5})

-7

min[[1 -3 7
-4 0 0.3]]

[-4 -3 0.3]

mirr()

Catalogo >

mirr

{

tassoFinanziamento

,*tassoReinvestimento*,*CF0*,*CFLista*
[,*CFFreq*])

Funzione finanziaria che restituisce l'indice interno di rendimento modificato di un investimento.

tassoFinanziamento è il tasso di interesse che si paga sugli importi di cash flow.

tassoReinvestimento è il tasso di interesse in corrispondenza del quale i cash flow vengono reinvestiti.

CF0 è il cash flow iniziale al tempo 0; deve essere un numero reale.

CFLista è una lista di importi di cash flow dopo il cash flow iniziale *CF0*.

CFFreq è una lista opzionale in cui ciascun elemento specifica la frequenza di occorrenza di un importo di cash flow raggruppato (consecutivo), che è l'elemento corrispondente di *CFLista*. L'impostazione predefinita è 1; è possibile inserire valori che siano numeri interi positivi < 10000.

Nota: vedere anche **irr()**, pagina 80.

list1:={6000,-8000,2000,-3000}

{6000, 8000,2000,-3000}

list2:={2,2,2,1}

{2,2,2,1}

mirr(4.65,12,5000,list1,list2)

13.41608607

mod() (Modulo)

Catalogo >

mod(Valore1, Valore2)⇒espressione

mod(Lista1, Lista2)⇒lista

mod(Matrice1, Matrice2)⇒matrice

Restituisce il primo argomento modulo secondo argomento, come definito dalle identità:

$$\text{mod}(x, 0) = x$$

$$\text{mod}(x, y) = x - y \text{ floor}(x/y)$$

Quando il secondo argomento è diverso da zero, il risultato è periodico in tale argomento. Il risultato può essere zero oppure ha lo stesso segno del secondo argomento.

Se gli argomenti sono due liste o due matrici, viene restituita una lista o una matrice contenente il modulo di ciascuna coppia di elementi corrispondenti.

Nota: vedere anche **remain()**, pagina 136

mod(7,0)	7
mod(7,3)	1
mod(-7,3)	2
mod(7,-3)	-2
mod(-7,-3)	-1
mod({12,-14,16},{9,7,-5})	{3,0,-4}

mRow() (Operazione con righe di matrice)

Catalogo >

mRow(Valore, Matrice1, Indice)⇒matrice

$$\text{mRow}\left(-1, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right) \quad \begin{bmatrix} 1 & 2 \\ -1 & \frac{-4}{3} \end{bmatrix}$$

Restituisce una copia di *Matrice1* dove ciascun elemento della riga *Indice* di *Matrice1* viene moltiplicato per *Valore*.

mRowAdd() (Moltiplicazione e somma di righe di matrice)

Catalogo >

mRowAdd(Valore, Matrice1, Indice1, Indice2)⇒matrice

$$\text{mRowAdd}\left(-3, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}$$

Restituisce una copia di *Matrice1* dove ciascun elemento della riga *Indice2* di *Matrice1* viene sostituito con:

$$\text{Valore} \times \text{riga } \text{Indice1} + \text{riga } \text{Indice2}$$

MultReg $Y, X1[, X2[, X3, \dots, X10]]$

Calcola la regressione lineare multipla della lista Y sulle liste $X1, X2, \dots, X10$. Il riepilogo dei risultati è memorizzato nella variabile $stat.results$. (pagina 160).

Tutte le liste devono avere la stessa dimensione.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $b0+b1 \cdot x1+b2 \cdot x2+ \dots$
stat.b0, stat.b1, ...	Coefficienti di regressione
stat.R ²	Coefficiente di determinazione multipla
stat.ŷList	\hat{y} List = $b0+b1 \cdot x1+ \dots$
stat.Resid	Residui della regressione

MultRegIntervals (Intervalli di confidenza della previsione di regressione multipla)

MultRegIntervals $Y, X1[, X2[, X3, \dots, X10]], listaValX[, livelloConfidenza]$

Calcola un valore y previsto, un intervallo di previsione del livello C per una singola osservazione e un intervallo di confidenza del livello C per la risposta media.

Il riepilogo dei risultati è memorizzato nella variabile $stat.results$.

(pagina 160).

Tutte le liste devono avere la stessa dimensione.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $b0+b1 \cdot x1+b2 \cdot x2+ \dots$
stat. \hat{y}	Una stima del punto: $\hat{y} = b0 + b1 \cdot x1 + \dots$ per <i>listaValX</i>
stat.dfError	Gradi di libertà degli errori
stat.CLower, stat.CUpper	Intervallo di confidenza per una risposta media
stat.ME	Margine di errore dell'intervallo di confidenza
stat.SE	Errore standard della risposta media
stat.LowerPred, stat.UpperrPred	Intervallo di previsione per una singola osservazione
stat.MEPred	Margine di errore dell'intervallo di previsione
stat.SEPred	Errore standard per la previsione
stat.bList	Lista dei coefficienti di regressione, {b0,b1,b2,...}
stat.Resid	Residui della regressione

MultRegTests (Verifica sulla regressione lineare multipla)

Catalogo > 

MultRegTests $Y, X1[, X2[, X3, \dots[, X10]]]$

Il test di regressione lineare multipla calcola una regressione lineare multipla sui dati assegnati e fornisce la statistica F test globale e le statistiche t test per i coefficienti.

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Output

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $b0+b1 \cdot x1+b2 \cdot x2+ \dots$
stat.F	Statistica della verifica F globale
stat.PVal	Valore P associato alla statistica F globale

Variabile di output	Descrizione
stat.R ²	Coefficiente di determinazione multipla
stat.AdjR ²	Coefficiente modificato di determinazione multipla
stat.s	Deviazione standard dell'errore
stat.DW	Statistica d Durbin-Watson; utilizzata per determinare se la correlazione automatica di primo ordine è presente nel modello
stat.dfReg	Gradi di liberà della regressione
stat.SSReg	Somma dei quadrati della regressione
stat.MSReg	Quadrato medio della regressione
stat.dfError	Gradi di libertà degli errori
stat.SSError	Somma dei quadrati degli errori
stat.MSError	Quadrato medio degli errori
stat.bList	{b0,b1,...} Lista dei coefficienti
stat.tList	Lista di statistiche t, una per ogni coefficiente di bList
stat.PList	Lista di valori P per ogni statistica t
stat.SEList	Lista di errori standard per coefficienti di bList
stat.yList	lista $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residui della regressione
stat.sResid	Residui standardizzati; valore ottenuto dividendo un residuo per la sua deviazione standard
stat.CookDist	Distanza di Cook; misura dell'influenza di un'osservazione basata sui valori residui e di leverage
stat.Leverage	Misura della distanza dei valori della variabile indipendente dai rispettivi valori medi

N

nand

ctrl **=** **tasti**

BooleanExpr1 nand BooleanExpr2
restuisce Boolean expression

BooleanList1 nand BooleanList2
restuisce Boolean list

BooleanMatrix1 nand BooleanMatrix2
restituisce Boolean matrix

Restituisce la negazione di un'operazione **and** logica riguardo i due argomenti.
Restituisce true, false o una forma semplificata dell'equazione.

Nel caso di liste e matrici, restituisce un confronto elemento per elemento.

Integer1 nand Integer2⇒integer

Confronta due interi reali bit per bit tramite un'operazione **nand**. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati i bit corrispondenti, il risultato sarà 0 se entrambi sono uguali a 1; in caso contrario, il risultato sarà 1. Il valore restituito rappresenta il risultato dei bit e viene visualizzato secondo la modalità base.

È possibile inserire gli interi in qualsiasi base numerica. Nel caso di un numero binario o esadecimale, è necessario utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli interi vengono considerati decimali (base 10)

3 and 4	0
3 nand 4	-1
{1,2,3} and {3,2,1}	{1,2,1}
{1,2,3} nand {3,2,1}	{-2,-3,-2}

nCr() (Combinazioni)

Catalogo >

nCr(Valore1, Valore2)⇒espressione

Dati i numeri interi *Valore1* e *Valore2* con *Valore1* ≥ *Valore2* ≥ 0, **nCr()** è il numero di combinazioni degli elementi di *Valore1* presi nel numero di *Valore2* per volta (questa procedura è nota anche come coefficiente binomiale).

nCr(Valore1, 0)⇒1

nCr(Valore1, interoNeg)⇒0

nCr(Valore1, interoPos)⇒ Valore1 · (Valore1-1)...(Valore1-interoPos+1)/

nCr(z,3) z=5	10
nCr(z,3) z=6	20

interoPos!

nCr(*Valore1, nonInter*)⇒*espressione1!*
 $((Valore1 - nonInter)!) \cdot nonInter!$

nCr(*Lista1, Lista2*)⇒*lista*

Restituisce una lista di combinazioni sulla base delle coppie di elementi corrispondenti nelle due liste. Le liste degli argomenti devono avere dimensioni uguali.

nCr(*Matrice1, Matrice2*)⇒*matrice*

Restituisce una matrice di combinazioni, sulla base delle coppie di elementi corrispondenti nelle due matrici. Le matrici degli argomenti devono avere le stesse dimensioni.

nCr({5,4,3},{2,4,2}) {10,1,3}

nCr[[6 5][4 3],[2 2][2 2]] [15 10][6 3]

nDerivative()

**nDerivative(*Espr1, Var=Valore*,
 $[, Ordine]$)**⇒*valore*

**nDerivative(*Espr1, Var[, Ordine]*) |
 $Var=Valore$** ⇒*valore*

Restituisce la derivata numerica calcolata con metodi di differenziazione automatica.

Se il *Valore* è specificato, lo stesso sostituisce qualsiasi assegnazione di variabile precedente o qualsiasi sovrapposizione corrente “|” della variabile.

Se la variabile *Var* non contiene un valore numerico, è necessario specificare *Valore*.

Ordine della derivata deve essere **1** o **2**.

Nota: L'algoritmo nDerivative() ha una limitazione: funziona in modo ricorsivo tramite l'espressione non semplificata, calcolando il valore numerico della derivata prima (e seconda, se applicabile) e di ciascuna sottoespressione, il che può produrre un risultato imprevisto.

nDerivative(x ,x=1)	1
nDerivative(x ,x) x=0	undef
nDerivative(sqrt(x-1),x) x=1	undef

nDerivative[x^(x^2+x)^3,x,1] x=0	undef
centralDiff[x^(x^2+x)^3,x] x=0	0.000033

nDerivative()

Catalogo >

Si consideri l'esempio sulla destra. La derivata prima di $x \cdot (x^2+x)^{1/3}$ con $x=0$ è uguale a 0. Tuttavia, poiché la derivata prima della sottoespressione $(x^2+x)^{1/3}$ è indefinita con $x=0$, e questo valore è utilizzato per calcolare la derivata dell'espressione totale, **nDerivative()** riporta il risultato come indefinito e visualizza un messaggio di avvertenza.

Se si incontra questa limitazione, verificare la soluzione graficamente. È anche possibile provare a utilizzare **centralDiff()**.

 newList() (Nuova lista)

Catalogo >

 newList(*numElementi*) \Rightarrow *lista*

newList(4) {0,0,0,0}

Restituisce una lista le cui dimensioni sono *numElementi*. Ciascun elemento è zero.

 newMat() (Nuova matrice)

Catalogo >

 **newMat(*numRighe*,
numColonne) \Rightarrow *matrice***newMat(2,3)
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Restituisce una matrice di zeri le cui dimensioni sono date da *numRighe* per *numColonne*.

 nfMax() (Massimo di una funzione calcolato numericamente)

Catalogo >

 nfMax(*Espr*, *Var*) \Rightarrow *valore*nfMax($-x^2 - 2 \cdot x - 1, x$) -1. **nfMax(*Espr*, *Var*, *estremoInf*) \Rightarrow *valore***nfMax($0.5 \cdot x^3 - x - 2, x, -5.5$) 5. **nfMax(*Espr*, *Var*, *estremoInf*,
estremoSup) \Rightarrow *valore***

**nfMax(*Espr*, *Var* | *estremoInf* \leq *Var*
 \leq *estremoSup*) \Rightarrow *valore***

Restituisce un possibile valore calcolato numericamente della variabile *Var* che produce il massimo locale di *Espr*.

nfMax() (Massimo di una funzione calcolato numericamente)

Catalogo >

Se si fornisce *estremoInf* e *estremoSup*, la funzione controlla nell'ambito dell'intervallo chiuso $[estremoInf, estremoSup]$ per il massimo locale.

nfMin() (Minimo di una funzione calcolato numericamente)

Catalogo >

nfMin(*Espr*, *Var*)⇒*valore*

$$\text{nfMin}(x^2 + 2 \cdot x + 5, x) \quad -1.$$

nfMin(*Espr*, *Var*, *estremoInf*)⇒*valore*

$$\text{nfMin}(0.5 \cdot x^3 - x - 2, x, -5, 5) \quad -5.$$

nfMin(*Espr*, *Var*, *estremoInf*, *estremoSup*)⇒*valore*

nfMin(*Espr*, *Var*) | *estremoInf*≤*Var*≤*estremoSup*⇒*valore*

Restituisce un possibile valore calcolato numericamente della variabile *Var* che produce il minimo locale di *Espr*.

Se si fornisce *estremoInf* e *estremoSup*, la funzione controlla nell'ambito dell'intervallo chiuso $[estremoInf, estremoSup]$ per il minimo locale.

nInt() (Integrale numerico)

Catalogo >

nInt(*Espr1*, *Var*, *Inferiore*, *Superiore*)⇒*espressione*

$$\text{nInt}(e^{-x^2}, x, -1, 1) \quad 1.49365$$

Se la funzione da integrare *Espr1* non contiene nessun'altra variabile oltre a *Var* e se *Inferiore* e *Superiore* sono costanti, ∞ positivo o ∞ negativo, allora **nInt()** restituisce un'approssimazione di $\int (Espr1, Var, Inferiore, Superiore)$. Tale approssimazione è una media pesata di alcuni valori esemplificativi della funzione da integrare nell'intervallo *Inferiore*<*Var*<*Superiore*.

nInt() (Integrale numerico)

Catalogo >

L'obiettivo sono sei cifre significative. L'algoritmo adattivo termina quando sembra che l'obiettivo sia stato raggiunto, oppure quando sembra che ulteriori esempi non potrebbero portare alcun miglioramento significativo.

Viene visualizzato un avvertimento ("Accuratezza dubbia") quando sembra che l'obiettivo non sia stato raggiunto.

È possibile nidificare nInt() per un'integrazione numerica multipla. I limiti di integrazione possono dipendere da variabili di integrazione esterne ad essi.

nInt($\cos(x), x, -\pi, \pi + 1.E-12$) -1.04144E-12

nInt(nInt($\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x$), x, 0, 1) 3.30423

nom()

Catalogo >

nom(*tassoEffettivo, CpY*) \Rightarrow valore

nom(5.90398,12) 5.75

Funzione finanziaria che converte il tasso di interesse effettivo annuo *tassoEffettivo* in un tasso nominale, essendo *CpY* il numero di periodi di capitalizzazione per anno.

tassoEffettivo deve essere un numero reale e *CpY* deve essere un numero reale > 0 .

Nota: vedere anche eff(), pagina 47.

nor

tasti

BooleanExpr1 nor *BooleanExpr2* restituisce
Boolean expression

BooleanList1 nor *BooleanList2* restituisce
Boolean list

BooleanMatrix1 nor *BooleanMatrix2*
restituisce *Boolean matrix*

Restituisce la negazione di un'operazione or logica riguardo i due argomenti. Restituisce true, false o una forma semplificata dell'equazione.

Nel caso di liste e matrici, restituisce un confronto elemento per elemento.

Integer1norInteger2⇒integer

Confronta due interi reali bit per bit tramite una operazione **nor**. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati i bit corrispondenti, il risultato sarà 1 se entrambi sono uguali a 1; in caso contrario, il risultato sarà 0. Il valore restituito rappresenta il risultato dei bit e viene visualizzato secondo la modalità base.

È possibile inserire gli integer in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli integer vengono considerati decimali (base 10)

3 or 4	7
3 nor 4	-8
{1,2,3} or {3,2,1}	{3,2,3}
{1,2,3} nor {3,2,1}	{-4,-3,-4}

norm() (Norma di Froebius)

Catalogo >

norm(Matrice)⇒espressione

norm(Vettore)⇒espressione

Restituisce la norma di Frobenius.

norm $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	5.47723
norm $\begin{bmatrix} 1 & 2 \end{bmatrix}$	2.23607
norm $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	2.23607

normCdf() (Probabilità di distribuzione normale)

Catalogo >

normCdf(valoreInferiore, valoreSuperiore [,μ[,σ]])⇒numero se *valoreInferiore* e *valoreSuperiore* sono numeri, *lista* se *valoreInferiore* e *valoreSuperiore* sono liste

Calcola la probabilità di distribuzione normale tra *valoreInferiore* e *valoreSuperiore* per μ (default=0) e σ (default=1) specificati.

Per $P(X \leq \text{valoreSuperiore})$, impostare *valoreInferiore* = -9E999.

normPdf($ValX$ [, μ [, σ]]) \Rightarrow numero se $ValX$ è un numero, lista se $ValX$ è una lista

Calcola la funzione della densità di probabilità per la distribuzione normale in un valore $valX$ specificato per μ σ specificati.

not

not *EsprBooleanal* \Rightarrow *espressione booleana*

Restituisce vero o falso o una forma semplificata dell'argomento.

not *Interol* \Rightarrow *intero*

Restituisce il complemento a uno di un intero reale. Internamente, *Inter01* viene convertito in numero binario a 64 bit con segno. Il valore di ciascun bit viene scambiato (0 diventa 1 e viceversa) per il complemento a uno. I risultati vengono visualizzati nella modalità Base che è stata impostata.

È possibile inserire l'intero in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, l'intero viene considerato decimale (base 10).

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Per ulteriori informazioni, vedere [Base2](#), pagina 17.

Catalogo > [a-z]

not (2≥3)	true
not 0hB0►Base16	0hFFFFFFFFFFFFFFF4F
not not 2	2

In modalità base Esadecimale:

Importante: è zero, non la lettera O.

not 0h7AC36 0hFFFFFFFFFFFF853C9

In modalità base Bin:

Per vedere l'intero risultato, premere ▲,
quindi utilizzare ◀ e ▶ per spostare il
cursore.

Nota: un numero binario può contenere fino a 64 cifre (oltre al prefisso 0b). Un numero esadecimale può contenere fino ad 16 cifre.

nPr() (Disposizioni semplici)**Catalogo >** **nPr(Valore1, Valore2)⇒espressione****nPr(Valore1, 0)⇒1****nPr(Valore1, interoNeg)⇒1/
((Valore1+1)·(Valore1+2)...
(Valore1-interoNeg))****nPr(Valore1, interoPos)⇒Valore1 ·
(Valore1-1)...(Valore1-interoPos+1)****nPr(Valore1, nonInter)⇒Valore1! /
(Valore1-nonInter)!****nPr(Lista1, Lista2)⇒lista**

Restituisce una lista delle disposizioni semplici sulla base delle coppie di elementi corrispondenti nelle due liste. Le liste degli argomenti devono avere dimensioni uguali.

nPr(Matrice1, Matrice2)⇒matrice

Restituisce una matrice di disposizioni semplici sulla base delle coppie di elementi corrispondenti nelle due matrici. Le matrici degli argomenti devono avere le stesse dimensioni.

nPr(z,3) z=5	60
nPr(z,3) z=6	120
nPr({5,4,3},{2,4,2})	{20,24,6}
nPr([6 5][4 3],[2 2][2 2])	[30 20] [12 6]

npv()**Catalogo >** **npv(tassoInteresse,CFO,CFLista
[,CFFreq])**

Funzione finanziaria che calcola il valore presente netto (Net Present Value), la somma dei valori presenti per i cash flow in entrata (somme ricevute) e in uscita (somme pagate). Un risultato positivo per npv indica un investimento proficuo.

tassoInteresse è il tasso a cui scontare i cash flow (il costo del denaro) di un dato periodo.

CF0 è il cash flow iniziale al tempo 0; deve essere un numero reale.

CFLista è una lista di importi di cash flow dopo il cash flow iniziale *CF0*.

nPr([6 5][4 3],[2 2][2 2])	[30 20] [12 6]
----------------------------	-------------------

list1:={6000,-8000,2000,-3000}	{6000, -8000, 2000, -3000}
list2:={2,2,2,1}	{2,2,2,1}
npv(10,5000,list1,list2)	4769.91

CFFreq è una lista in cui ciascun elemento specifica la frequenza di occorrenza di un importo di cash flow raggruppato (consecutivo), che è l'elemento corrispondente di *CFLista*. L'impostazione predefinita è 1; è possibile inserire valori che siano numeri interi positivi < 10000.

nSolve() (Soluzione numerica)

nSolve(*Equazione*,*Var*[=*Campione*]) ⇒ numero o stringa_erroro

nSolve(*Equazione*,*Var*[=*Campione*],*estremoInf*) ⇒ numero o stringa_erroro

nSolve(*Equazione*,*Var*[=*Campione*],*estremoInf*,*estremoSup*) ⇒ numero o stringa_erroro

nSolve(*Equazione*,*Var*[=*Campione*]) | *estremoInf*≤*Var*≤*estremoSup* ⇒ numero o stringa_erroro

Ricerca iterativamente una soluzione numerica reale approssimata dell'*Equazione* per la sua variabile. Specificare la variabile come:

variabile

– o –

variabile = numero reale

Ad esempio, x è valido come pure x=3.

nSolve() cerca di determinare un punto in cui il resto sia zero oppure due punti relativamente vicini, nei quali il resto abbia segni opposti e la grandezza del resto non sia eccessiva. Se non è possibile ottenere ciò utilizzando un numero limitato di punti campione, viene restituita la stringa "Nessuna soluzione trovata."

nSolve($x^2+5 \cdot x - 25 = 9, x$)	3.84429
nSolve($x^2 = 4, x = -1$)	-2.
nSolve($x^2 = 4, x = 1$)	2.

Nota: se vi sono soluzioni multiple, è possibile usare un valore campione per trovare una soluzione particolare.

nSolve($x^2+5 \cdot x - 25 = 9, x$) $x < 0$	-8.84429
nSolve($\frac{(1+r)^{24}-1}{r} = 26, r$) $r > 0$ and $r < 0.25$	0.006886
nSolve($x^2 = 1, x$)	"No solution found"

OneVar (Statistiche a una variabile)**Catalogo > ****OneVar [1,]X,[Freq],[Categoria,Includi]]****OneVar [n,]X1,X2[X3,[...,[X20]]]**

Calcola le statistiche ad una variabile fino a 20 liste. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

Gli argomenti di X sono liste di dati.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici di categoria dei dati corrispondenti di *X*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Un elemento vuoto (nullo) in qualsiasi lista *X*, *Freq* o *Categoria* produce un corrispondente elemento vuoto in tutte queste liste. Un elemento vuoto (nullo) in qualsiasi lista da *X1* a *X20* produce un corrispondente elemento vuoto in tutte queste liste. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

Variabile di output	Descrizione
stat. \bar{x}	Media dei valori X
Statistiche. Σx	Somma dei valori X
stat. Σx^2	Somma dei valori x^2
stat.sx	Deviazione standard del campione di X

Variabile di output	Descrizione
stat.x	Deviazione standard della popolazione di X
stat.n	Numero dei punti di dati
stat.MinX	Minimo dei valori x
stat.Q ₁ X	1° quartile di x
stat.MedianX	Mediana di x
stat.Q ₃ X	3° quartile di x
stat.MaxX	Massimo dei valori x
stat.SSX	Somma dei quadrati delle deviazioni dalla media di x

or

Catalogo >

BooleanExpr1 or *BooleanExpr2* restituisce
Boolean expression

BooleanList1 or *BooleanList2* restituisce
Boolean list

BooleanMatrix1 or *BooleanMatrix2*
restituisce Boolean matrix

Restituisce vero o falso o una forma
semplificata dell'espressione immessa
originariamente.

Restituisce vero se la semplificazione di
una o di entrambe le espressioni risulta
vera. Restituisce falso solo se il calcolo di
entrambe le espressioni risulta falso.

Nota: vedere xor.

Nota per l'inserimento dell'esempio: per
istruzioni sull'inserimento di definizioni di
programmi e funzioni costituite da più
righe, consultare la sezione Calcolatrice del
manuale del prodotto.

Intero1 or *Intero2* \Rightarrow *intero*

Define $g(x) = \begin{cases} \text{Func} & \text{If } x \leq 0 \text{ or } x \geq 5 \\ \text{If } x \leq 0 \text{ or } x \geq 5 \\ \text{Goto end} \\ \text{Return } x \cdot 3 \\ \text{Lbl end} \\ \text{EndFunc} \end{cases}$

$g(3)$	9
$g(0)$	A function did not return a value

9

A function did not return a value

In modalità base Esadecimale:

0h7AC36 or 0h3D5F	0h7BD7F
-------------------	---------

Importante: è zero, non la lettera O.

Confronta due interi reali bit per bit tramite un'operazione or. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati bit corrispondenti, il risultato sarà 1 se uno dei due bit è uguale a 1; se entrambi i bit sono uguali a 0, il risultato sarà 0. Il valore restituito rappresenta il risultato dei bit e viene visualizzato nella modalità base che è stata impostata.

È possibile inserire gli interi in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli interi vengono considerati decimali (base 10).

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Per ulteriori informazioni, vedere **►Base2**, pagina 17.

Nota: vedere xor.

In modalità base Bin:

0b100101 or 0b100	0b100101
-------------------	----------

Nota: un numero binario può contenere fino a 64 cifre (oltre al prefisso 0b). Un numero esadecimale può contenere fino ad 16 cifre.

ord() Codice numerico di carattere

ord(Stringa)⇒intero

ord(Lista)⇒lista

Restituisce il codice numerico del primo carattere nella stringa di caratteri *Stringa*, oppure una lista dei primi caratteri di ciascun elemento della lista.

ord("hello")	104
char(104)	"h"
ord(char(24))	24
ord({ "alpha", "beta" })	{ 97,98 }

P

►Rx() (Coordinata x rettangolare)

►Rx(rEspr, θEspr)⇒espressione

►Rx(rLista, θLista)⇒lista

►Rx(rMatrice, θMatrice)⇒matrice

In modalità angolo in radianti:

►Rx(4,60°)	2.
►Rx({ -3,10,1.3 }, { $\frac{\pi}{3}$, $\frac{-\pi}{4}$, 0 })	{ -1.5,7.07107,1.3 }

Restituisce la coordinata x equivalente della coppia (r, θ).

Nota: l'argomento θ viene interpretato come angolo in gradi, gradianti o radianti, conformemente alla modalità di misurazione degli angoli impostata. Se l'argomento è un'espressione, si può utilizzare il simbolo °, G o r per escludere tale impostazione provvisoriamente.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **P@>Rx (...)**.

▶Ry() (Coordinata y rettangolare)

▶Ry(rValore, θValore)⇒valore

In modalità angolo in radianti:

▶Ry(rLista, θLista)⇒lista

P▶Ry(4,60°) 3.4641

▶Ry(rMatrice, θMatrice)⇒matrice

P▶Ry({{-3,10,1,3}},{{π/3, -π/4, 0}})
 $\{-2.59808, -7.07107, 0\}$

Restituisce la coordinata y equivalente della coppia (r, θ).

Nota: l'argomento θ viene interpretato come angolo in gradi, gradianti o radianti, conformemente alla modalità di misurazione degli angoli impostata.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **P@>Ry (...)**.

PassErr

PassErr

Per un esempio di **PassErr**, vedere l'esempio 2 del comando **Try**, pagina 172.

Passa un errore al livello successivo.

Se la variabile di sistema *errCode* è zero, **PassErr** non esegue alcuna azione.

L'istruzione **Else** del blocco **Try...Else...EndTry** dovrebbe utilizzare **ClrErr** o **PassErr**. Se l'errore deve essere elaborato o ignorato, utilizzare **ClrErr**. Se non si sa quale azione applicare all'errore, utilizzare **PassErr** per inviarlo al successivo blocco di gestione degli errori. Se non ci sono ulteriori blocchi di gestione degli errori **Try...Else...EndTry** in attesa di applicazione, la finestra di dialogo dell'errore viene visualizzata come normale.

Nota: Controlla anche **ClrErr**, pagina 24, e **Try**, pagina 172.

Nota per l'inserzione dell'esempio:

Nell'applicazione Calcolatrice sul palmare, è possibile inserire definizioni su più linee premendo **[enter]** invece di **[enter]** alla fine di ogni riga. Sulla tastiera del computer, mantenere premuto **Alt** e premere **Invio**.

piecewise() (Funzione definita a tratti)

piecewise(*Espr1* [, *Condizione1* [, *Espr2* [, *Condizione2* [, ...]]]])

Restituisce definizioni di una funzione piecewise (definita a tratti) sotto forma di elenco. È inoltre possibile creare definizioni piecewise utilizzando un modello.

Nota: vedere anche **Modello di funzione piecewise** a pagina 3.

Define $p(x)=\begin{cases} x, & x>0 \\ \text{undef}, & x\leq 0 \end{cases}$	Done
$p(1)$	1
$p(-1)$	undef

poissCdf() (Probabilità cumulata per la distribuzione discreta di Poisson)

```
poissCdf
(
λ
,
valoreInferiore,valoreSuperiore)⇒numero
se valoreInferiore e valoreSuperiore sono
numeri, lista se valoreInferiore e
valoreSuperiore sono liste
```

poissCdf(λ ,*valoreSuperiore*)(per $P(0 \leq X \leq \text{valoreSuperiore}) \Rightarrow \text{numero}$ se

poissCdf() (Probabilità cumulata per la distribuzione discreta di Poisson)**Catalogo > ►**

valoreSuperiore è un numero, lista se
valoreSuperiore è una lista

Calcola una probabilità cumulata per la distribuzione discreta di Poisson con la media λ specificata.

Per $P(X \leq valoreSuperiore)$, impostare
valoreInferiore=0

poissPdf() (Probabilità per la distribuzione discreta di Poisson)**Catalogo > ►**

poissPdf(λ , ValX) \Rightarrow numero se *ValX* è un numero, lista se *ValX* è una lista

Calcola una probabilità per la distribuzione discreta di Poisson con la media λ specificata.

►Polar (Visualizza come vettore polare)**Catalogo > ►**

Vettore ►Polar

[1 3.] ►Polar

[3.16228 ∠ 71.5651]

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Polar.

Visualizza *vettore* in forma polare $[r \angle \theta]$. Il vettore deve essere bidimensionale e può essere sia una riga che una colonna.

Nota: ►Polar è un'istruzione in formato di visualizzazione, non una funzione di conversione. Tale istruzione può essere impiegata solo alla fine di una riga di introduzione e non aggiorna *ans*.

Nota: vedere anche ►Rect, pagina 133.

valoreComplesso ►Polar

In modalità angolo in radianti:

Visualizza *vettoreComplesso* in forma polare.

- In modalità angolo in gradi, restituisce $(r \angle \theta)$.
- In modalità angolo in radianti, restituisce $r e^{i\theta}$.

$(3+4 \cdot i)$ ►Polar	$e^{0.927295 \cdot i \cdot 5}$
$\left(4 \angle \frac{\pi}{3}\right)$ ►Polar	$e^{1.0472 \cdot i \cdot 4}$

►Polar (Visualizza come vettore polare)

Catalogo >

valoreComplesso può avere una forma complessa qualsiasi. Tuttavia una voce $re^{i\theta}$ causa un errore nella modalità di misurazione degli angoli in gradi.

Nota: è necessario usare le parentesi per un inserimento polare ($r \angle \theta$).

In modalità angolo in gradi (gradi centesimali):

($4 \cdot i$)►Polar

($4 \angle 100$)

In modalità angolo in gradi:

($3+4 \cdot i$)►Polar

($5 \angle 53.1301$)

polyEval() (Calcola polinomio)

Catalogo >

polyEval(Lista1, Espr1)⇒espressione

polyEval([1,2,3,4],2) 26

polyEval(Lista1, Lista2)⇒espressione

polyEval([1,2,3,4],[2,-7]) {26,-262}

Interpreta il primo argomento come coefficiente di un polinomio di grado decrescente e restituisce il polinomio calcolato per il valore del secondo argomento.

polyRoots()

Catalogo >

polyRoots(Poli,Var)⇒lista

polyRoots(y^3+1,y) {-1}

polyRoots(ListaDiCoeff)⇒lista

cPolyRoots(y^3+1,y) {-1,0.5-0.866025i,0.5+0.866025i}

La prima sintassi, **polyRoots(Poli,Var)**, restituisce una lista di radici reali del polinomio *Poli* in funzione della variabile *Var*. Se non esistono radici reali, restituisce una lista vuota: {}.

polyRoots($x^2+2 \cdot x+1,x$) {-1,-1}

Poli deve essere un polinomio in forma normale in una variabile. Non utilizzare polinomi non ridotti a forma normale, quali $y^2 \cdot y+1$ o $x \cdot x+2 \cdot x+1$

polyRoots({1,2,1}) {-1,-1}

La seconda sintassi, **PolyRoots** (*ListaDiCoeff*), restituisce una lista di radici reali per i coefficienti di *ListaDiCoeff*.

Nota: vedere anche **cPolyRoots()**, pagina 33.

PowerReg *X,Y[,Freq][,Categoria,
Includi]]*

Calcola la regressione su potenzay = (a · (x)^b) sulle liste X e Y con frequenza Freq. Il riepilogo dei risultati è memorizzato nella variabile stat.results. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di Includi.

X e Y sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di Freq specifica la frequenza di occorrenza di ogni dato corrispondente di X e Y. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti X e Y..

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: a · (x) ^b
stat.a, stat.b	Coefficienti di regressione
stat.r ²	Coefficiente di determinazione lineare di dati trasformati
stat.r	Coefficiente di correlazione per dati trasformati ($\ln(x)$, $\ln(y)$)
stat.Resid	Residui associati al modello di potenza
stat.ResidTrans	Residui associati all'adattamento lineare dei dati trasformati
stat.XReg	Lista di punti dati della Lista X modificata attualmente usata nella regressione secondo le restrizioni di Freq, Lista Categoria e Includi Categorie

Variabile di output	Descrizione
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

Prgm

Catalogo >

Prgm
Blocco
EndPrgm

Modello per la creazione di un programma definito dall'utente. Deve essere utilizzato con il comando **Define**, **Define LibPub** o **Define LibPriv**.

Blocco può essere una singola istruzione, una serie di istruzioni separate dal carattere ":" o una serie di istruzioni su righe separate.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Calcolare il massimo comune divisore e visualizzare i risultati intermedi.

Define $\text{proggcd}(a,b)=\text{Prgm}$
 Local d
 While $b \neq 0$
 $d:=\text{mod}(a,b)$
 $a:=b$
 $b:=d$
 Disp $a, " ", b$
 EndWhile
 Disp "GCD=", a
 EndPrgm

Done

$\text{proggcd}(4560,450)$

450 60
60 30
30 0
GCD=30

Done

prodSeq()

Vedere **P()**, pagina 204.

Product (PI) (Prodotto)

Vedere **P()**, pagina 204.

product() (Prodotto)

Catalogo >

product(*Lista*[, *Inizio*[, *Fine*]]) \Rightarrow *espressione*

$\text{product}(\{1,2,3,4\})$	24
$\text{product}(\{4,5,8,9\},2,3)$	40

product() (Prodotto)

Catalogo >

Restituisce il prodotto degli elementi contenuti in *Lista*. *Inizio* e *Fine* sono opzionali e specificano un intervallo di elementi.

product(*MatriceI*[, *Inizio*[, *Fine*]]) \Rightarrow *matrice*

Restituisce un vettore di riga contenente i prodotti degli elementi nelle colonne di *MatriceI*. *Inizio* e *Fine* sono opzionali e specificano un intervallo di elementi.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

product	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	[28 80 162]
product	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, 1..2$	[4 10 18]

propFrac() (Frazione propria)

Catalogo >

propFrac(*ValoreI*[, *Var*]) \Rightarrow *valore*

propFrac(*numero_razionale*) restituisce *numero_razionale* sotto forma di somma di un numero intero e di una frazione, aventi lo stesso segno e denominatore di grandezza maggiore del numeratore.

propFrac(*espressione_razionale*, *Var*) restituisce la somma delle frazioni proprie ed un polinomio rispetto a *Var*. Il grado di *Var* nel denominatore supera il grado di *Var* nel numeratore di ciascuna frazione propria. Le potenze simili di *Var* sono ridotte. I termini ed i rispettivi fattori sono ordinati considerando *Var* la variabile principale.

Se *Var* è omesso, viene eseguita l'espansione della frazione propria rispetto alla variabile più significativa. I coefficienti della parte polinomiale vengono trasformati in propri rispetto alla prima variabile più significativa, e così di seguito.

propFrac	$1\frac{1}{3}$	$1 + \frac{1}{3}$
propFrac	$-1\frac{1}{3}$	$-1 - \frac{1}{3}$

QR (Scomposizione QR)**Catalogo >** **QR Matrice, MatriceQ, MatriceR[, Tol]**

Calcola la scomposizione QR di Householder di una matrice reale o complessa. Le matrici Q ed R che si ottengono vengono memorizzate nei *Matrice* specificati. La matrice Q è unitaria. La matrice R è triangolare superiore.

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tol*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tol* viene ignorato.

- Se si usa **ctrl enter** oppure se si imposta la modalità **Auto o Approssimato** su Approssimato, i calcoli verranno eseguiti in virgola mobile.
- Se *Tol* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:

$$5E-14 \cdot \max(\dim(\text{Matrice})) \cdot \text{rowNorm}(\text{Matrice})$$

La scomposizione QR viene calcolata in modo numerico tramite trasformazioni di Householder; la soluzione simbolica tramite Gram-Schmidt. Le colonne in *nomeMatriceQ* sono i vettori della base ortonormale con estensione pari allo spazio definito da *matrice*.

Il numero a virgola mobile (9.) in *m1* fa sì che i risultati vengano calcolati nella forma a virgola mobile.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
QR <i>m1,qm,rm</i>	Done
<i>qm</i>	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

QuadReg (Regressione quadratica)**Catalogo >** **QuadReg *X,Y[, Freq] [, Categoria, Includi]***

Calcola la regressione polinomiale quadraticay = a · x²+b · x+c sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Coefficienti di regressione
stat.R ²	coefficiente di determinazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

QuartReg *X, Y [, Freq] [, Categoria, Includi]*

Calcola la regressione polinomiale quarticay
 $= a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$ sulle liste *X* e *Y*
 con frequenza *Freq*. Il riepilogo dei risultati
 è memorizzato nella variabile *stat.results*.
 (pagina 160).

Tutte le liste devono avere le stesse
 dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e
 dipendenti.

Freq è una lista opzionale di valori di
 frequenza. Ciascun elemento di *Freq*
 specifica la frequenza di occorrenza di ogni
 dato corrispondente di *X* e *Y*. Il valore
 predefinito è 1. Tutti gli elementi devono
 essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di
 stringa di categoria dei dati corrispondenti *X*
 e *Y*.

Includi è una lista di uno o più codici di
 categoria. Solo quei dati il cui codice di
 categoria è inserito in questa lista vengono
 inclusi nel calcolo.

Per informazioni sull'effetto di elementi
 vuoti in una lista, vedere "Elementi vuoti
 (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Coefficienti di regressione
stat.R ²	coefficiente di determinazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

R►Pθ()

R►Pθ (valoreX, valoreY) ⇒ valore
R►Pθ (listaX, listaY) ⇒ lista
R►Pθ (matriceX, matriceY) ⇒ matrice

Restituisce la coordinata θ equivalente della coppia di argomenti (x,y) .

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradi centesimali o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **R@>Ptheta (...).**

Catalogo > [R]

In modalità angolo in gradi:

R►Pθ(2,2) 45.

In modalità angolo in gradi centesimali:

R►Pθ(2,2) 50.

In modalità angolo in radianti:

R►Pθ(3,2) 0.588003

R►Pθ([3 -4 2], [0 $\frac{\pi}{4}$ 1.5])
 $[0. 2.94771 \quad 0.643501]$

R►Pr()

R►Pr (valoreX, valoreY) ⇒ valore
R►Pr (listaX, listaY) ⇒ lista
R►Pr (matriceX, matriceY) ⇒ matrice

Restituisce la coordinata r equivalente alla coppia di argomenti (x,y) .

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **R@>Pr (...).**

Catalogo > [R]

In modalità angolo in radianti:

R►Pr(3,2) 3.60555

R►Pr([3 -4 2], [0 $\frac{\pi}{4}$ 1.5])
 $[3. 4.07638 \quad \frac{5}{2}]$

► Rad

Valore1►Rad ⇒ valore

Converte l'argomento in una misura d'angolo in radianti.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando **@>Rad.**

Catalogo > [R]

In modalità angolo in gradi:

(1.5)►Rad $(0.02618)^r$

In modalità angolo in gradi centesimali:

(1.5)►Rad $(0.023562)^r$

rand() (Numero casuale)

Catalogo >

rand() \Rightarrow espressione

rand(*numTentativi*) \Rightarrow lista

rand() restituisce un numero casuale compreso tra 0 e 1.

rand(*numTentativi*) restituisce una lista contenente *numTentativi* valori casuali compresi tra 0 e 1.

Imposta il seme dei numeri casuali.

RandSeed 1147	Done
rand(2)	{ 0.158206, 0.717917 }

randBin() (Numero casuale da distribuzione binomiale)

Catalogo >

randBin(*n, p*) \Rightarrow espressione

randBin(*n, p, numTentativi*) \Rightarrow lista

randBin(*n, p*) restituisce un numero reale casuale da una distribuzione binomiale specificata.

randBin(*n, p, numTentativi*) restituisce una lista contenente *numTentativi* numeri reali casuali da una distribuzione binomiale specificata.

randBin(80,0.5)	46.
randBin(80,0.5,3)	{ 43., 39., 41. }

randInt() (Intero casuale)

Catalogo >

randInt

(

estremoInf,
estremoSup) \Rightarrow

espressione

randInt

(

estremoInf,

estremoSup

,*numTentativi*) \Rightarrow

lista

randInt(3,10) 3.

randInt(3,10,4) { 9., 3., 4., 7. }

randInt() (Intero casuale)

Catalogo >

randInt
(
estremoInf,
estremoSup)
restituisce un intero
casuale all'interno
dell'intervallo
specificato dai
numeri interi
estremoInf e
estremoSup.

randInt
(
estremoInf,
'
estremoSup,
numTentativi)
restituisce un elenco
contenente
numTentativi interi
casuali all'interno
dell'intervallo
specificato.

randMat() (Matrice casuale)

Catalogo >

randMat(*numRighe*, *numColonne*) \Rightarrow
matrice

Restituisce una matrice di numeri interi
compresi tra -9 e 9 della dimensione
specificata.

Entrambi gli argomenti devono potere
essere semplificati in numeri interi.

RandSeed 1147	Done
randMat(3,3)	$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$

Nota: i valori di questa matrice cambiano
ogni volta che si preme **enter**.

randNorm() (Normale casuale)

Catalogo >

randNorm(μ , σ) \Rightarrow *espressione*
randNorm(μ , σ , *numTentativi*) \Rightarrow *lista*

RandSeed 1147	Done
randNorm(0,1)	0.492541
randNorm(3,4.5)	-3.54356

randNorm() (Normale casuale)**Catalogo > **

randNorm(μ, σ) restituisce un numero decimale dalla distribuzione normale specificata. Può essere qualsiasi numero reale, anche se con maggiore probabilità sarà compreso nell'intervallo $[\mu - 3\sigma, \mu + 3\sigma]$.

randNorm($\mu, \sigma, numTentativi$) restituisce una lista contenente $numTentativi$ valori decimali dalla distribuzione normale specificata.

randPoly() (Polinomio casuale)**Catalogo > **

randPoly($Var, Ordine$) \Rightarrow espressione

Restituisce un polinomio in Var dell'*Ordine* specificato. I coefficienti sono interi casuali compresi tra -9 e 9. Il primo coefficiente non può essere zero.

Ordine deve essere tra 0 e 99.

RandSeed 1147	Done
randPoly($x, 5$)	$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$

randSamp() (Campione casuale)**Catalogo > **

randSamp([Lista, numTentativi [, noSostituzione]]) \Rightarrow lista

Restituisce una lista contenente un campione casuale di *numTentativi* da *Lista* con l'opzione di sostituire il campione (*noSostituzione*=0) o meno (*noSostituzione*=1). L'impostazione predefinita prevede la sostituzione del campione.

Define <i>list3</i> = $\{1,2,3,4,5\}$	Done
Define <i>list4</i> =randSamp(<i>list3, 6</i>)	Done
<i>list4</i>	$\{1.,3.,3.,1.,3.,1.\}$

RandSeed (Seme numero casuale)**Catalogo > **

RandSeed Numero

Se *Numero* = 0, imposta i semi ai valori predefiniti per il generatore di numeri casuali. Se *Numero* \neq 0, viene utilizzato per generare due semi, memorizzati nelle variabili di sistema seed1 e seed2.

RandSeed 1147	Done
rand()	0.158206

real() (Reale)**Catalogo >** **real(ValoreI) ⇒ valore**real($2+3 \cdot i$)

2

Restituisce la parte reale dell'argomento.

real(Listal) ⇒ listareal({ $1+3 \cdot i, 3, i$ })

{1,3,0}

Restituisce le parti reali di tutti gli elementi.

real(MatriceI) ⇒ matricereal($\begin{bmatrix} 1+3 \cdot i & 3 \\ 2 & i \end{bmatrix}$) $\begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix}$

Restituisce le parti reali di tutti gli elementi.

► Rect**Catalogo >** **Vettore ► Rect****Nota:** è possibile inserire questo operatore dalla tastiera del computer digitando @>Rect. $\left\{ 3 \angle \frac{\pi}{4} \angle \frac{\pi}{6} \right\} \blacktriangleright \text{Rect}$

[1.06066 1.06066 2.59808]

Visualizza *Vettore* nella forma rettangolare [x, y, z]. Le dimensioni del vettore devono essere 2 o 3 e il vettore può essere una riga o una colonna.**Nota:** ►Rect è un'istruzione del formato di visualizzazione, non una funzione di conversione. Tale istruzione può essere impiegata solo alla fine di una riga di introduzione e non aggiorna ans.**Nota:** Vedere anche ►Polar, pagina 121.**valoreComplesso ► Rect**Visualizza *valoreComplesso* nella forma rettangolare a+bi. *valoreComplesso* può avere una forma complessa qualsiasi. Tuttavia una voce $re^{i\theta}$ causa un errore nella modalità di misurazione degli angoli in gradi.**Nota:** è necessario usare le parentesi per un inserimento polare ($r\angle\theta$).

In modalità angolo in radianti:

 $\left\{ 4 \cdot e^{\frac{\pi}{3}} \right\} \blacktriangleright \text{Rect}$

11.3986

 $\left\{ 4 \angle \frac{\pi}{3} \right\} \blacktriangleright \text{Rect}$

2.+3.4641·i

In modalità angolo in gradi centesimali:

 $\left\{ (1 \angle 100) \right\} \blacktriangleright \text{Rect}$

i

In modalità angolo in gradi:

 $\left\{ (4 \angle 60) \right\} \blacktriangleright \text{Rect}$

2.+3.4641·i

Nota: per inserire \angle dalla tastiera, selezionarlo nell'elenco dei simboli del Catalogo.

ref() (Forma a scalini per righe)

Catalogo >

ref(*Matrice1[, Tol]*) \Rightarrow *matrice*

Restituisce la forma a scalini per righe di *Matrice1*.

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tol*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tol* viene ignorato.

$$\text{ref} \begin{pmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{pmatrix} \quad \begin{pmatrix} 1 & -\frac{2}{5} & -\frac{4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & -\frac{62}{71} \end{pmatrix}$$

- Se si usa **[ctrl] [enter]** oppure se si imposta la modalità **Auto o Approssimato** su Approssimato, i calcoli saranno eseguiti in virgola mobile.
- Se *Tol* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:
 $5E-14 * \max(\dim(\text{Matrice1})) * \text{rowNorm}(\text{Matrice1})$

Evitare elementi indefiniti in *Matrice1*.

Possono produrre risultati imprevisti.

Ad esempio, se *a* è indefinito nella seguente espressione, viene visualizzato un messaggio di avvertenza e il risultato viene mostrato come:

$$\text{ref} \begin{pmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Il messaggio di avvertenza appare perché l'elemento generalizzato $1/a$ non è valido per $a=0$.

È possibile evitare questa situazione memorizzando in anticipo un valore in *a* oppure utilizzando l'operatore di vincolo ("|"), come mostrato nell'esempio che segue.

$$\text{ref}\left[\begin{matrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\right] | a=0 \quad \left[\begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{matrix}\right]$$

Nota: vedere anche rref(), page 144.

RefreshProbeVars

RefreshProbeVars

Permette di accedere ai dati di tutti i sensori collegati in un programma TI-Basic.

Valore di varStato Stato

varStato Normale (continuare l'esecuzione
=0 del programma)
L'applicazione Vernier DataQuest™ è in modalità di raccolta dati.

varStato **Nota:** affinché questo comando funzioni, l'applicazione Vernier DataQuest™ deve essere in
=1 modalità misura. 

varStato L'applicazione Vernier DataQuest™
=2 non è stata avviata.

varStato L'applicazione Vernier DataQuest™
=3 è stata avviata, ma non sono presenti sensori collegati.

Esempio

```
Define temp ()=
Prgm
    © Verificare se il sistema è pronto
    RefreshProbeVars stato
    If stato=0 Then
        Disp "pronto"
        Per n,1,50
        RefreshProbeVars stato
        temperatura:=meter.temperatura
        Disp "Temperatura: ",temperatura
        If temperatura>30 Then
            Disp "Caldo eccessivo"
        EndIf
        © Attendere 1 secondo tra i campioni
        Wait 1
    EndFor
Else
```

Disp "Non pronto. Riprovare più tardi"

EndIf

EndPrgm

Nota: Può essere utilizzato anche con l'hub di TI-Innovator™.

remain() (Resto)

Catalogo >

remain(Valore1, Valore2) ⇒ valore
remain(Lista1, Lista2) ⇒ lista
remain(Matrice1, Matrice2) ⇒ matrice

Restituisce il resto del primo argomento rispetto al secondo argomento, come definito dalle identità:

remain(x,0) x
 remain(x,y) x - y•iPart(x/y)

Si noti che, di conseguenza **remain(-x,y) = -remain(x,y)**. Il risultato può essere zero oppure può avere lo stesso segno del primo argomento.

Nota: vedere anche **mod()**, pagina 103.

remain(7,0)	7
remain(7,3)	1
remain(-7,3)	-1
remain(7,-3)	1
remain(-7, -3)	-1
remain({12, -14, 16}, {9, 7, -5})	{3, 0, 1}

$$\text{remain}\begin{bmatrix} 9 & -7 \\ 6 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 3 \\ 4 & -3 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$$

Request

Catalogo >

Request stringaPrompt, var[, FlagDisp [, varState]]

Request promptString, funz(arg1, ...argn) [, FlagDisp [, statusVar]]

Istruzione di programmazione: Sospende il programma e visualizza una finestra di dialogo contenente il messaggio *stringaPrompt* e un riquadro di testo per la risposta dell'utente.

Dopo che l'utente ha digitato una risposta e ha fatto clic su **OK**, il contenuto del riquadro di testo viene assegnato alla variabile *var*.

Definire un programma:

```
Define request_demo()=Prgm
  Request "Radius: ",r
  Disp "Area = ",pi*r^2
EndPrgm
```

Eseguire il programma e digitare una risposta:

request_demo()

Se l'utente fa clic su **Cancel** (Annulla), il programma procede senza accettare input. Il programma utilizza il precedente valore di *var* se *var* è già stata definita.

L'argomento opzionale *FlagDispl* può essere qualsiasi espressione.

- Se *FlagDispl* viene omesso o dà come risultato **1**, la richiesta e la risposta da parte dell'utente vengono visualizzate nella cronologia della calcolatrice.
- Se *FlagDispl* dà come risultato **0**, la richiesta e la risposta non vengono visualizzate nella cronologia.

L'argomento *varStato* opzionale offre al programma la possibilità di determinare la modalità di uscita dell'utente dalla finestra di dialogo. Notare che *varStato* richiede l'argomento *FlagDispl*.

- Se l'utente ha fatto clic su **OK** o ha premuto **Invio** oppure **Ctrl+Invio**, la variabile *varStato* viene impostata sul valore **1**.
- Altrimenti, la variabile *varStato* viene impostata sul valore **0**.

L'argomento *funz()* consente a un programma di memorizzare la risposta dell'utente come una definizione di funzione. Questa sintassi opera come se l'utente avesse eseguito il comando:

Define funz(arg1, ...argn) = risposta utente

Il programma può quindi usare la funzione definita *funz()*. La *stringaPrompt* dovrebbe portare l'utente ad inserire una *risposta utente* appropriata che completa la definizione di funzione.

Nota: è possibile utilizzare il comando *Request* all'interno di un programma definito dall'utente, ma non di una funzione.



Risultato dopo aver selezionato **OK**:

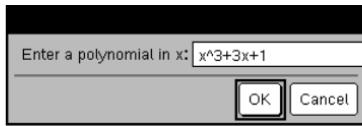
Semiretta: 6/2
Area = 28.2743

Definire un programma:

```
Define polynomial()=Prgm
  Request "Enter a polynomial in
x:",p(x)
  Disp "Real roots are:",polyRoots
  (p(x),x)
EndPrgm
```

Eseguire il programma e digitare una risposta:

polynomial()



Risultato dopo avere inserito x^3+3x+1 e selezionato **OK**:

Le radici reali sono: {-0.322185}

Per arrestare un programma che contiene un comando **Request** in un loop infinito:

- **Palmare:** Tenere premuto il tasto  **on** e premere **enter** più volte.
- **Windows®:** Tenere premuto il tasto **F12** e premere **Invio** più volte.
- **Macintosh®:** Tenere premuto il tasto **F5** e premere **Invio** più volte.
- **iPad®:** L'app mostra un prompt. È possibile continuare ad attendere o annullare.

Nota: vedere anche **RequestStr**, page 138.

RequestStr

RequestStr *stringaPrompt*, *var*[, *DispFlag*]

Istruzione di programmazione: Opera in modo identico alla prima sintassi del comando **Request**, eccetto che la risposta dell'utente viene sempre interpretata come stringa. A differenza di questo, il comando **Request** interpreta la risposta come espressione a meno che l'utente non la racchiuda tra virgolette ("").

Nota: è possibile utilizzare il comando **RequestStr** all'interno di un programma definito dall'utente, ma non di una funzione.

Per arrestare un programma che contiene un comando **RequestStr** in un loop infinito:

- **Palmare:** Tenere premuto il tasto  **on** e premere **enter** più volte.
- **Windows®:** Tenere premuto il tasto **F12** e premere **Invio** più volte.
- **Macintosh®:** Tenere premuto il tasto **F5** e premere **Invio** più volte.
- **iPad®:** L'app mostra un prompt. È possibile continuare ad attendere o annullare.

Definire un programma:

```
Define requestStr_demo()=Prgm
  RequestStr "Your name:",name,0
  Disp "Response has ",dim(name)," characters."
EndPrgm
```

Eseguire il programma e digitare una risposta:

```
requestStr_demo()
```



Risultato dopo aver selezionato **OK** (si noti che l'argomento *FlagDisp* 0 omette la richiesta e la risposta dalla cronologia):

```
requestStr_demo()
```

La risposta ha 5 caratteri.

Nota: vedere anche **Request**, page 136.

Return**Return [Espr]**

Restituisce *Espr* quale risultato della funzione. Questo comando viene utilizzato all'interno di un blocco **Func...EndFunc**.

Nota: utilizzare **Return** senza alcun argomento all'interno di un blocco **Prgm...EndPrgm** per uscire da un programma.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define **factorial (nn)=**

```
Func
Local answer,counter
1 → answer
For counter,1,nn
answer·counter → answer
EndFor
Return answer
EndFunc
```

factorial (3)

6

right() (Destra)**right(ListaI[, Num])** ⇒ *lista*

right{ {1,3,-2,4},3 }

{3,-2,4}

Restituisce i *Num* elementi più a destra contenuti in *ListaI*.

Se si omette *Num*, restituisce l'intera *ListaI*.

right(stringaOrigine[, Num]) ⇒ *stringa*

right"Hello",2

"lo"

Restituisce i caratteri *Num* più a destra contenuti nella stringa di caratteri *stringaOrigine*.

Se si omette *Num*, restituisce l'intera *stringaOrigine*.

right(Confronto) ⇒ *espressione*

Restituisce il secondo membro di un'equazione o di una disequazione.

rk23 ()**rk23(Espr, Var, varDipendente, {Var0, VarMax}, varDipendente0, incrVar [,**

Equazione differenziale:

tollErrore]) \Rightarrow matrice

rk23(*SistemaDiEspr, Var, ListaDiVarDipendenti, {Var0, VarMax}, ListaDiVarDipendenti0, incrVar[, tollErrore])* \Rightarrow matrice

rk23(*ListaDiEspr, Var, ListaDiVarDipendenti, {Var0, VarMax}, ListaDiVarDipendenti0, incrVar[, tollErrore])* \Rightarrow matrice

Utilizza il metodo di Runge-Kutta per risolvere il sistema

$$\frac{d \ depVar}{d \ Var} = Expr(Var, depVar)$$

con *varDipendente*

(Var0)=varDipendente0 nell'intervallo [*Var0,VarMax*]. Restituisce una matrice la cui prima riga definisce i valori ottenuti di *Var* come definito da *incrVar*. La seconda riga definisce il valore del primo componente della soluzione per i valori di *Var* corrispondenti, e così via.

Espr è il secondo membro che definisce l'equazione differenziale ordinaria (ODE).

SistemaDiEspr è un sistema di secondi membri che definiscono il sistema di ODE (corrisponde all'ordine di variabili dipendenti in *ListaDivarDipendenti*).

ListaDiEspr è una lista di secondi membri che definisce il sistema di ODE (corrisponde all'ordine di variabili dipendenti in *ListaDiVarDipendenti*).

Var è la variabile indipendente.

ListaDiVarDipendenti è una lista di variabili dipendenti.

{*Var0, VarMax*} è una lista a due elementi che indica alla funzione di integrare da *Var0* a *VarMax*.

ListaDiVarDipendenti0 è una lista di valori iniziali di variabili dipendenti.

$$y'=0.001*y*(100-y) \text{ e } y(0)=10$$

$$\begin{aligned} \text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right) \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9367 & 11.9493 & 13.042 & 14.2 \end{bmatrix} \end{aligned}$$

Per vedere l'intero risultato, premere \blacktriangleleft , quindi utilizzare \blacktriangleleft e \blacktriangleright per spostare il cursore.

Stessa equazione con *tollErrore* impostata su **1.E-6**

$$\begin{aligned} \text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1, 1.E-6\right) \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9367 & 11.9495 & 13.0423 & 14.2189 \end{bmatrix} \end{aligned}$$

Sistema di equazioni

$$\begin{cases} y1' = y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

con $y1(0)=2$ e $y2(0)=5$

$$\begin{aligned} \text{rk23}\left(\begin{cases} y1' = 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right) \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 2. & 1.94103 & 4.78694 & 3.25253 & 1.82848 \\ 5. & 16.8311 & 12.3133 & 3.51112 & 6.27245 \end{bmatrix} \end{aligned}$$

Se $incrVar$ dà come risultato un numero diverso da zero: $\text{sign}(incrVar) = \text{sign}(VarMax - Var0)$ e sono restituite soluzioni a $Var0 + i * incrVar$ per tutti i valori di $i = 0, 1, 2, \dots$ tali che $Var0 + i * incrVar$ sia in $[var0, VarMax]$ (potrebbe non esserci un valore di soluzione a $VarMax$).

Se *incrVar* dà come risultato zero, le soluzioni sono restituite ai valori *Var* di "Runge-Kutta".

tollErrore è la tolleranza d'errore (predefinita a 0.001).

root() (Radice)

Catalogo > 

root(*Valore*) \Rightarrow radice

root(Valore1, Valore2) \Rightarrow radice

root(*Valore*) restituisce la radice quadrata di *Valore*.

$$\sqrt[3]{8}$$

2

3

144225

root(*Valore1*, *Valore2*) restituisce la radice *Valore2* di *Valore1*. *Valore1* può essere una costante reale o complessa a virgola mobile, una costante razionale intera o complessa.

Nota: Vedere anche **Modello di radice ennesima a pagina, pagina 1.**

rotate() (Ruota)

Catalogo >

rotate(*Interol*[, *numRotazioni*]) \Rightarrow *intero*

Ruota i bit di un numero intero binario. È possibile inserire *Interol* in qualsiasi base numerica; esso viene convertito automaticamente in forma binaria a 64 bit con segno. Se *Interol* è troppo grande per questa forma, un'operazione a modulo simmetrico lo porta all'interno dell'intervallo. Per ulteriori informazioni, vedere ► **Base2**, pagina 17.

In modalità base Bin:

Per vedere l'intero risultato, premere ▲,
quindi utilizzare ◀ e ▶ per spostare il
cursore.

rotate() (Ruota)

Catalogo >

Se *numRotazioni* è positivo, la rotazione avviene verso sinistra. Se *numRotazioni* è negativo, la rotazione avviene verso destra. L'impostazione predefinita è -1 (ruota a destra di un bit).

Ad esempio, in una rotazione a destra:

Ogni bit ruota verso destra.

0b0000000000000001111010110000110101

L'ultimo bit a destra diventa il primo a sinistra.

dà:

0b100000000000000111101011000011010

I risultati vengono visualizzati nella modalità Base che è stata impostata.

rotate(Lista1[,numRotazioni]) ⇒ *lista*

Restituisce una copia di *Lista1* ruotata a destra o a sinistra di *numRotazioni* elementi. Non modifica *Lista1*.

Se *numRotazioni* è positivo, la rotazione avviene verso sinistra. Se *numRotazioni* è negativo, la rotazione avviene verso destra. L'impostazione predefinita è -1 (ruota a destra di un bit).

rotate(Stringa1[,numRotazioni]) ⇒ *stringa*

Restituisce una copia di *Stringa1* ruotata a destra o a sinistra di *numRotazioni* caratteri. Non modifica *Stringa1*.

Se *numRotazioni* è positivo, la rotazione avviene verso sinistra. Se *numRotazioni* è negativo, la rotazione avviene verso destra. L'impostazione predefinita è -1 (ruota a destra di un carattere).

In modalità base Esadecimale:

rotate(0h78E)	0h3C7
rotate(0h78E,-2)	0h8000000000000001E3
rotate(0h78E,2)	0h1E38

Importante: per inserire un numero binario o esadecimale, utilizzare sempre il prefisso 0b o 0h (è uno zero, non la lettera O).

In modalità base Dec:

rotate({1,2,3,4})	{4,1,2,3}
rotate({1,2,3,4},-2)	{3,4,1,2}
rotate({1,2,3,4},1)	{2,3,4,1}

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"beda"

round() (Arrotondamento)

Catalogo >

round(Valore1[, cifre]) ⇒ *valore*

round(1.234567,3)	1.235
-------------------	-------

round() (Arrotondamento)

Catalogo >

Restituisce l'argomento arrotondato ad un numero specifico di cifre dopo la virgola decimale.

cifre deve essere un numero intero compreso tra 0 e 12. Se *cifre* non è incluso, restituisce l'argomento arrotondato alle prime 12 cifre significative.

Nota: la visualizzazione dipende dalla modalità selezionata.

round(Lista1[, cifre]) \Rightarrow lista

Restituisce una lista degli elementi arrotondati al numero specifico di cifre.

$$\text{round}(\{\pi, \sqrt{2}, \ln(2)\}, 4) \\ \{3.1416, 1.4142, 0.6931\}$$

round(Matrice1[, cifre]) \Rightarrow matrice

Restituisce una matrice degli elementi arrotondati al numero specifico di cifre.

$$\text{round}\left(\begin{bmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{bmatrix}, 1\right) \\ \begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$$

rowAdd() (Somma di righe di matrice)

Catalogo >

rowAdd(Matrice1, rIndice1, rIndice2) \Rightarrow matrice

Restituisce una copia di *Matrice1* nella quale la riga *rIndice2* è sostituita dalla somma delle righe *rIndice1* e *rIndice2*.

$$\text{rowAdd}\left(\begin{bmatrix} 3 & 4 \\ -3 & -2 \end{bmatrix}, 1, 2\right) \\ \begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$$

rowDim() (Dimensione righe matrice)

Catalogo >

rowDim(Matrice) \Rightarrow espressione

Restituisce il numero di righe di *Matrice*.

Nota: vedere anche **colDim()**, pagina 25.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1 \\ \text{rowDim}(m1) \\ \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \\ 3$$

rowNorm() (Norma righe matrice)

Catalogo >

rowNorm(Matrice) \Rightarrow espressione

Restituisce il massimo delle somme dei valori assoluti degli elementi nelle righe di *Matrice*.

$$\text{rowNorm}\left(\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix}\right) \\ 25$$

Nota: tutti gli elementi della matrice devono potere essere semplificati in numeri. Vedere anche **colNorm()**, pagina 25.

rowSwap() (Inverti righe matrice)

rowSwap(*Matrice1*, *rIndice1*, *rIndice2*) \Rightarrow *matrice*

Restituisce *Matrice1* con le righe *rIndice1* e *rIndice2* scambiate.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
rowSwap(<i>mat</i> , 1, 3)	$\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$

rref() (Forma a scalini ridotta per righe)

rref(*Matrice1*[, *Tol*]) \Rightarrow *matrice*

Restituisce la forma a scalini ridotta per righe di *Matrice1*.

$\text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$
--	---

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tol*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tol* viene ignorato.

- Se si usa **ctrl enter** oppure se si imposta la modalità **Auto o Approssimato** su Approssimato, i calcoli saranno eseguiti in virgola mobile.
- Se *Tol* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:

$$5E-14 \cdot \max(\dim(\text{Matrice1})) \cdot \text{rowNorm}(\text{Matrice1})$$

Nota: vedere anche **ref()**, page 134.

sec() (Secante)**Tasto**

$\text{sec}(\text{Valore}1) \Rightarrow \text{valore}$

$\text{sec}(\text{Listal}) \Rightarrow \text{lista}$

Restituisce la secante di *Valore1* oppure restituisce una lista contenente le secanti di tutti gli elementi in *Listal*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, l'argomento viene interpretato come angolo in gradi, gradianti o radianti. È possibile utilizzare °, G o r per escludere provvisoriamente la modalità d'angolo selezionata.

sec⁻¹() (Secante inversa)**Tasto**

$\text{sec}^{-1}(\text{Valore}1) \Rightarrow \text{valore}$

$\text{sec}^{-1}(\text{Listal}) \Rightarrow \text{lista}$

Restituisce l'angolo la cui secante è *Valore1* oppure restituisce una lista contenente le secanti inverse di ciascun elemento di *Listal*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradianti o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arcsec (...)**.

In modalità angolo in gradi:

$\text{sec}(45) \quad 1.41421$

$\text{sec}(\{1,2,3,4\}) \quad \{1.00015, 1.00081, 1.00244\}$

In modalità angolo in gradi:

$\text{sec}^{-1}(1) \quad 0.$

In modalità angolo in gradi (gradi centesimali):

$\text{sec}^{-1}(\sqrt{2}) \quad 50.$

In modalità angolo in radianti:

$\text{sec}^{-1}(\{1,2,5\}) \quad \{0,1.0472, 1.36944\}$

sech() (Secante iperbolica)**Catalogo** >

$\text{sech}(\text{Valore}1) \Rightarrow \text{valore}$

$\text{sech}(\text{Listal}) \Rightarrow \text{lista}$

Restituisce la secante iperbolica di *Valore1* oppure restituisce una lista contenente le secanti iperboliche di tutti gli elementi di *Listal*.

$\text{sech}(3) \quad 0.099328$

$\text{sech}(\{1,2,3,4\}) \quad \{0.648054, 0.198522, 0.036619\}$

sech⁻¹(Valore1) ⇒ valore**sech⁻¹(Lista1) ⇒ lista**

Restituisce la secante iperbolica inversa di *Valore1* oppure restituisce una lista contenente le secanti iperboliche inverse di ciascun elemento di *Lista1*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arcsech(...)**.

In modalità angolo in radianti e in modalità formato rettangolare complesso:

sech ⁻¹ (1)	0
sech ⁻¹ { { 1,-2,2,1 } }	{ 0,2.0944·i,8.e-15+1.07448·i }

Send

Menu Hub

Send exprOrString1[, exprOrString2] ...

Istruzione di programmazione: invia uno o più TI-Innovator™ Hub comandi a un hub collegato.

exprOrString deve essere un TI-Innovator™ Hub comando valido. Normalmente, *exprOrString* contiene un comando "SET ..." per controllare un dispositivo o un comando "READ ..." per richiedere i dati.

Gli argomenti vengono inviati in successione all'hub.

Nota: è possibile utilizzare il comando **Send** all'interno di un programma definito dall'utente ma non di una funzione.

Nota: vedere anche **Get** (pagina 64), **GetStr** (pagina 71) e **eval()** (pagina 51).

Esempio: attivare l'elemento blu del LED RGB integrato per 0,5 secondi.

Send "SET COLOR.BLUE ON TIME .5"	Done
----------------------------------	------

Esempio: richiedere il valore attuale del sensore di livello luminosità integrato dell'hub. Un comando **Get** recupera il valore e lo assegna alla variabile *lightval*.

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	0.347922

Esempio: inviare una frequenza calcolata al diffusore integrato dell'hub. Utilizzare la variabile speciale *iostr.SendAns* per mostrare il comando hub con l'espressione valutata.

<i>n</i> :=50	50
<i>m</i> :=4	4
Send "SET SOUND eval(<i>m</i> · <i>n</i>)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

seq() (Sequenza)

Catalogo >

seq(*Espr, Var, Basso, Alto[, Incr]*) \Rightarrow lista

Aumenta *Var* da *Basso* a *Alto* con incrementi di *Incr*, calcola *Espr* e restituisce i risultati in forma di lista. Il contenuto originale di *Var* è intatto dopo l'esecuzione di **seq()**.

Il valore predefinito di *Incr* è 1.

$\text{seq}\left(n^2, n, 1, 6\right)$	$\{1, 4, 9, 16, 25, 36\}$
$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$	$\left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	$\frac{1968329}{1270080}$

Nota: Per forzare un risultato approssimativo,

Palmare: Premere **ctrl** **enter**.

Windows®: Premere **Ctrl+Invio**.

Macintosh®: Premere **⌘+Invio**.

iPad®: Tenere premuto **Invio** e selezionare **≈**.

$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	1.54977
--	---------

seqGen()

Catalogo >

seqGen(*Espr, Var, varDipendente, {Var0, VarMax}[], ListaDiTermIniziali [, incrVar [, ValArrotPerEccesso]]*) \Rightarrow lista

Genera una lista di termini per la successione *varDipendente*(*Var*)=*Espr* come segue: Incrementa la variabile indipendente *Var* da *Var0* fino a *VarMax* di *incrVar*, calcola *varDipendente*(*Var*) per i valori corrispondenti di *Var* utilizzando la formula *Espr* e *ListaDiTermIniziali* e restituisce i risultati in forma di lista.

seqGen(*ListaOSistemaDiEspr, Var, ListaDiVarDipendenti, {Var0, VarMax}[], MatriceDiTermIniziali [, incrVar [, ValArrotPerEccesso]]*) \Rightarrow matrice

Generare i primi 5 termini della sequenza $u(n) = u(n-1)^2/2$, con $u(1)=2$ e *incrVar*=1.

$\text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right)$	$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$
---	---

Esempio in cui *Var0*=2:

$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right)$	$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$
---	--

Sistema di due successioni:

seqGen()

Catalogo >

Genera una matrice di termini per un sistema (o lista) di successioni

ListaDiVarDipendenti

(*Var*)=*ListaOSistemaDiEspr* come segue:
Incrementa la variabile indipendente *Var* da *Var0* fino a *VarMax* di *incrVar*, calcola *ListaDiVarDipendenti(Var)* per i valori corrispondenti di *Var* utilizzando la formula *ListaOSistemaDiEspr* e *MatriceDiTermIniziali* e restituisce i risultati in forma di matrice.

Il contenuto originale di *Var* è intatto dopo l'esecuzione di **seqGen()**.

Il valore predefinito di *incrVar* è **1**.

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u2(n-1)}{2} + u1(n-1)\right\}, n, \{u1, u2\}, \{1, 5\}, \begin{bmatrix} - \\ 2 \end{bmatrix}\right)$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix}$$

Nota: il simbolo di nullo (-) nella matrice di termini iniziali sopra viene utilizzato per indicare che il termine iniziale per $u1(n)$ è calcolato utilizzando la formula di sequenza esplicita $u1(n)=1/n$.

seqn()

Catalogo >

seqn(*Espr{u, n [, ListaDiTermIniziali[, nMax [, ValArrotPerEccesso]]}]*) \Rightarrow *lista*

Genera una lista di termini per una successione $u(n)=\text{Espr}(u, n)$ come segue: incrementa *n* da 1 a *nMax* di 1, calcola $u(n)$ per i valori di *n* corrispondenti utilizzando la formula *Espr(u, n)* e *ListaDiTermIniziali* e restituisce i risultati in forma di lista.

seqn(*Espr{n [, nMax [, ValArrotPerEccesso]}]*) \Rightarrow *lista*

Genera una lista di termini per una successione non ricorsiva $u(n)=\text{Espr}(n)$ come segue: Incrementa *n* da 1 a *nMax* di 1, calcola $u(n)$ per i valori di *n* corrispondenti utilizzando la formula *Espr(n)* e restituisce i risultati in forma di lista.

Se manca *nMax*, *nMax* viene impostata su 2500

Se *nMax*=0, *nMax* viene impostato su 2500

Nota: **seqn()** chiama **seqGen()** con *n0*=1 e *nincr*=1

Genera i primi 6 termini della successione $u(n)=u(n-1)/2$, con $u(1)=2$.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{ 2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360} \right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{ 1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36} \right\}$$

**setMode(*interoNomeModo*,
interoImpostazione)** \Rightarrow *intero*

setMode(*lista*) \Rightarrow *lista interi*

Valido solo all'interno di una funzione o di un programma.

**setMode(*interoNomeModo*,
interoImpostazione)** imposta temporaneamente il modo *interoNomeModo* sulla nuova impostazione *interoImpostazione* e restituisce un intero corrispondente all'impostazione originale di quel modo. La modifica è limitata alla durata dell'esecuzione del programma o della funzione.

interoNomeModo specifica il modo da impostare. Deve essere uno degli interi dei modi della tabella seguente.

interoImpostazione specifica la nuova impostazione per il modo. Deve essere uno dei numeri interi per le impostazioni elencati di seguito per lo specifico modo che si sta impostando.

setMode(*lista*) consente di modificare impostazioni multiple. *lista* contiene coppie di interi dei modi e di interi delle impostazioni. **setMode(*lista*)** restituisce una lista simile le cui coppie di interi rappresentano i modi e le impostazioni originali.

Se si salvano tutte le impostazioni di modo con **getMode(0) \rightarrow var**, è possibile utilizzare **setMode(var)** per ripristinare tali impostazioni fintantoché la funzione o il programma esistono. Vedere **getMode()**, pagina 70.

Visualizzare il valore approssimato di π utilizzando l'impostazione predefinita di Mostra cifre (Display Digits), quindi visualizzare π con l'impostazione Fissa2. Verificare che l'impostazione predefinita venga ripristinata dopo l'esecuzione del programma.

Define <i>prog1()</i> =Prgm	Done
Disp π	
setMode(1,16)	
Disp π	
EndPrgm	
<i>prog1()</i>	
	3.14159
	3.14
	Done

Nota: le impostazioni di modo correnti vengono passate alle sottoroutine chiamate. Se una sottoroutine cambia un'impostazione di modo, tale modifica viene perduta quando il controllo torna alla routine di chiamata.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Nome modo	Intero modo	Interi impostazioni
Mostra cifre (Display digits)	1	1 =Mobile, 2 =Mobile1, 3 =Mobile2, 4 =Mobile3, 5 =Mobile4, 6 =Mobile5, 7 =Mobile6, 8 =Mobile7, 9 =Mobile8, 10 =Mobile9, 11 =Mobile10, 12 =Mobile11, 13 =Mobile12, 14 =Fissa0, 15 =Fissa1, 16 =Fissa2, 17 =Fissa3, 18 =Fissa4, 19 =Fissa5, 20 =Fissa6, 21 =Fissa7, 22 =Fissa8, 23 =Fissa9, 24 =Fissa10, 25 =Fissa11, 26 =Fissa12
Angolo (Angle)	2	1 =Radiani, 2 =Gradi, 3 =Gradanti
Formato esponenziale (Exponential Format)	3	1 =Normale, 2 =Scientifico, 3 =Tecnico
Reale o Complesso (Real or Complex)	4	1 =Reale, 2 =Rettangolare, 3 =Polare
Auto o Approssimato (Auto or Approx.)	5	1 =Auto, 2 =Approssimato
Formato vettoriale (Vector Format)	6	1 =Rettangolare, 2 =Cilindrico, 3 =Sferico
Base	7	1 =Decimale, 2 =Esa, 3 =Binario

shift(Intero1[,numSpostamenti])⇒intero

Sposta i bit di un numero intero binario. È possibile inserire *Intero1* in qualsiasi base numerica; esso viene convertito automaticamente in forma binaria a 64 bit con segno. Se *Intero1* è troppo grande per questa forma, un'operazione a modulo simmetrico lo porta all'interno dell'intervallo. Per ulteriori informazioni, vedere **►Base2**, pagina 17.

Se *numSpostamenti* è positivo, lo spostamento avviene verso sinistra. Se *numSpostamenti* è negativo, lo spostamento avviene verso destra. L'impostazione predefinita è -1 (sposta a destra di un bit).

In uno spostamento a destra, il bit più a destra viene eliminato e al suo posto viene inserito 0 o 1, in modo che corrisponda al bit più a sinistra. In uno spostamento a sinistra, il bit più a sinistra viene eliminato e viene inserito 0 come bit più a destra.

Ad esempio, in uno spostamento a destra:

ogni bit viene spostato verso destra.

0b000000000000000111101011000011010

Inserisce 0 se il bit più a sinistra è 0,

oppure 1 se il bit più a sinistra è 1.

Dà:

0b000000000000000111101011000011010

I risultati vengono visualizzati nella modalità Base che è stata impostata. Gli zeri iniziali non vengono visualizzati.

shift(Lista1 [,numSpostamenti])⇒lista

Restituisce una copia di *Lista1* spostata a destra o a sinistra di *numSpostamenti* elementi. Non modifica *Lista1*.

In modalità base Bin:

shift(0b1111010110000110101)	0b1111010110000110101
shift(256,1)	0b1000000000

In modalità base Esadecimale:

shift(0h78E)	0h3C7
shift(0h78E,-2)	0h1E3
shift(0h78E,2)	0h1E38

Importante: per inserire un numero binario o esadecimale, utilizzare sempre il prefisso 0b o 0h (è uno zero, non la lettera O).

In modalità base Dec:

shift({1,2,3,4})	{ undef,1,2,3 }
shift({1,2,3,4},-2)	{ undef,undef,1,2 }
shift({1,2,3,4},2)	{ 3,4,undef,undef }

Se *numSpostamenti* è positivo, lo spostamento avviene verso sinistra. Se *numSpostamenti* è negativo, lo spostamento avviene verso destra. L'impostazione predefinita è -1 (sposta a destra di un elemento).

Gli elementi introdotti all'inizio o alla fine di *lista* a seguito dello spostamento sono contrassegnati con il simbolo "undef".

**shift(*Stringa1*,
[,*numSpostamenti*])**⇒*stringa*

Restituisce una copia di *Stringa1* spostata a sinistra o a destra di *numSpostamenti* caratteri. Non modifica *Stringa1*.

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

Se *numSpostamenti* è positivo, lo spostamento avviene verso sinistra. Se *numSpostamenti* è negativo, lo spostamento avviene verso destra. L'impostazione predefinita è -1 (sposta a destra di un carattere).

I caratteri introdotti all'inizio o alla fine di *stringa* a seguito dello spostamento sono costituiti da uno spazio.

sign() (Segno)

sign(*Valore1*)⇒*valore*

sign(-3.2)	-1
sign({2,3,4,5})	{1,1,1,-1}

sign(*Listal*)⇒*lista*

sign(*Matrice1*)⇒*matrice*

Quando *Valore1* è reale o complesso, restituisce *Valore1* / **abs(Valore1)** se *Valore1* ≠ 0.

Restituisce 1 se *Valore1* è positivo.

Restituisce -1 se *Valore1* è negativo.

sign(0) restituisce ±1 se la modalità Formato complesso è Reale; altrimenti restituisce se stesso.

sign(0) rappresenta la circonferenza unitaria del dominio complesso.

Se la modalità del formato complesso è Reale:

sign([-3 0 3])	[-1 undef 1]
----------------	--------------

Con liste o matrici, restituisce i segni di tutti gli elementi.

simult() Sistema di equazioni

simultanee

**simult(*matriceCoeff*, *vettoreCost*,
[,*Tol*])**⇒*matrice*

Restituisce un vettore colonna che contiene le soluzioni di un sistema di equazioni lineari.

Nota: vedere anche **linSolve()**, pagina 89.

matriceCoeff deve essere una matrice quadrata contenente i coefficienti delle equazioni.

vettoreCost deve avere lo stesso numero di righe (stesse dimensioni) di *matriceCoeff* e deve contenere i termini noti.

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tol*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tol* viene ignorato.

- Se si imposta la modalità **Auto o Approssimato** su Approssimato, i calcoli verranno eseguiti in virgola mobile.
- Se *Tol* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come: $5E-14 \cdot \max(\dim(\text{matriceCoeff})) \cdot \text{rowNorm}(\text{matriceCoeff})$

**simult(*matriceCoeff*, *matriceCost*[,
]*Tol*])**⇒*matrice*

Risolve sistemi multipli di equazioni lineari in cui ogni sistema ha coefficienti delle equazioni uguali ma termini noti diversi.

Risolvere rispetto a x e y:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$\begin{array}{|c|c|} \hline \text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) & \begin{bmatrix} -3 \\ 2 \end{bmatrix} \\ \hline \end{array}$$

La soluzione è x=-3 e y=2.

Risolvere:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{array}{|c|c|} \hline \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \text{matx1} & \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\ \hline \text{simult}\left(\text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) & \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \\ \hline \end{array}$$

Risolvere:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

simult() Sistema di equazioni simultanee

Catalogo >

Ogni colonna di *matriceCost* deve contenere i termini noti per un sistema di equazioni. Ogni colonna della matrice risultante contiene la soluzione per il sistema corrispondente.

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right) = \begin{bmatrix} -3 & -7 \\ 2 & 9 \\ 2 & 2 \end{bmatrix}$$

Per il primo sistema, $x=-3$ e $y=2$. Per il secondo sistema, $x=-7$ e $y=9/2$.

sin() (Seno)

sin(ValoreI)⇒valore

sin(ListaI)⇒lista

sin(ValoreI) restituisce il seno dell'argomento.

sin(ListaI) restituisce una lista di seni di tutti gli elementi di *ListaI*.

Nota: l'argomento è interpretato come angolo in gradi, gradianti o radianti, a seconda della modalità angolo correntemente impostata. Si può usare ${}^{\circ}$, G o r per escludere tale impostazione provvisoriamente.

Tasto

In modalità angolo in gradi:

$\sin\left(\frac{\pi}{4}\right)_r$	0.707107
$\sin(45)$	0.707107
$\sin\{0,60,90\}$	{0.,0.866025,1.}

In modalità angolo in gradi centesimali:

$\sin(50)$	0.707107
------------	----------

In modalità angolo in radianti:

$\sin\left(\frac{\pi}{4}\right)$	0.707107
$\sin(45^{\circ})$	0.707107

In modalità angolo in radianti:

$\sin\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$
--	--

sin(matriceQuadrataI)⇒matriceQuadrata

Restituisce il seno della matrice di *matriceQuadrataI*. Ciò non equivale a calcolare il seno di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrataI deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

Tasto

sin⁻¹⁽⁾ (Arcoseno)

sin⁻¹(ValoreI)⇒valore

In modalità angolo in gradi:

sin⁻¹() (Arcoseno)

Tasto 

sin⁻¹(Listal)⇒lista

sin⁻¹(1)

90.

sin⁻¹(Valore1) restituisce l'angolo il cui seno è *Valore1*.

sin⁻¹(Lista1) restituisce una lista contenente l'inversa del seno di ciascun elemento di *Lista1*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradianti o radienti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arcsin(...)**.

sin⁻¹(matriceQuadrata1)⇒matriceQuadrata

Restituisce l'inversa del seno della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare l'inversa del seno di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in gradianti (gradi centesimali):

sin⁻¹(1)

100.

In modalità angolo in radianti:

sin⁻¹({0,0.2,0.5}) { 0.,0.201358,0.523599 }

In modalità angolo in radianti e in modalità formato rettangolare complesso:

sin⁻¹($\begin{pmatrix} 1 & 5 \\ 4 & 2 \end{pmatrix}$)
[-0.174533-0.12198·i 1.74533-2.35591·i]
[1.39626-1.88473·i 0.174533-0.593162·i]

sinh() (Seno iperbolico)

Catalogo > 

sinh(Valore1)⇒valore

sinh(1.2) 1.50946

sinh(Listal)⇒lista

sinh({0,1,2,3.}) { 0,1.50946,10.0179 }

sinh(Valore1) restituisce il seno iperbolico dell'argomento.

sinh(Listal) restituisce una lista del seno iperbolico di ciascun elemento di *Lista1*.

sinh(matriceQuadrata1)⇒matriceQuadrata

Restituisce il seno iperbolico della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il seno iperbolico di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

In modalità angolo in radianti:

sinh($\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$)
[360.954 305.708 239.604]
[352.912 233.495 193.564]
[298.632 154.599 140.251]

sinh() (Seno iperbolico)

Catalogo >

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

sinh⁻¹() (Arcoseno iperbolico)

Catalogo >

sinh⁻¹(Valore1)⇒valore

$\sinh^{-1}(0)$ 0

sinh⁻¹(Listal)⇒lista

$\sinh^{-1}(\{0, 0.2, 1, 3\})$ {0, 0.148748, 1.81845}

sinh⁻¹(Valore1) restituisce l'inversa del seno iperbolico dell'argomento.

sinh⁻¹(Listl) restituisce una lista contenente l'inversa del seno iperbolico di ciascun elemento di *Listal*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arcsinh(...)**.

sinh¹
(matriceQuadrata1)⇒matriceQuadrata

Restituisce l'inversa del seno iperbolico della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare l'inversa del seno iperbolico di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

In modalità angolo in radianti:

$\sinh^{-1}\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$
[0.041751 2.15557 1.1582]
1.46382 0.926568 0.112557
2.75079 -1.5283 0.57268

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

SinReg (Regressione sinusoidale)

Catalogo >

SinReg X, Y [, [Iterazioni], [Periodo] [, Categoria, Includi]]

Calcola la regressione sinusoidale sulle liste *X* e *Y*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Iterazioni è un valore che specifica quante volte al massimo (da 1 a 6) verrà tentata una soluzione. Se omesso, viene utilizzato 8. Di solito valori più alti danno una maggiore accuratezza ma richiedono tempi di esecuzione più lunghi, e viceversa.

Periodo indica il periodo stimato. Se viene omesso, la differenza tra i valori di *X* deve essere uguale e in ordine sequenziale. Se invece *Periodo* viene specificato, le differenze tra valori x possono non essere uguali.

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

L'output di **SinReg** è sempre espresso in radianti, indipendentemente dall'impostazione corrente della modalità di misurazione degli angoli.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Coefficienti di regressione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

SortA (Ordinamento ascendente)

Catalogo >

SortA *List1[, Lista2] [, Lista3] ...*

SortA *Vettore1[, Vettore2] [, Vettore3] ...*

Questo comando permette di ordinare in modo ascendente gli elementi del primo argomento.

Se vengono inclusi altri argomenti, il comando permette di ordinare gli elementi di ciascuno di essi in modo che le loro nuove posizioni coincidano con le nuove posizioni degli elementi del primo argomento.

Tutti gli argomenti devono essere nomi di liste o di vettori e devono avere le stesse dimensioni.

Gli elementi vuoti (nulli) nel primo argomento si spostano in fondo. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
SortA <i>list1</i>	<i>Done</i>
<i>list1</i>	$\{1,2,3,4\}$
$\{4,3,2,1\} \rightarrow list2$	$\{4,3,2,1\}$
SortA <i>list2,list1</i>	<i>Done</i>
<i>list2</i>	$\{1,2,3,4\}$
<i>list1</i>	$\{4,3,2,1\}$

SortD (Ordinamento discendente)

Catalogo >

SortD *List1[, Lista2] [, Lista3] ...*

SortD *Vettore1[, Vettore2] [, Vettore3] ...*

Questo comando è identico a **SortA**, con la differenza che **SortD** ordina gli elementi in modo discendente.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
SortD <i>list1,list2</i>	<i>Done</i>
<i>list1</i>	$\{4,3,2,1\}$
<i>list2</i>	$\{3,4,1,2\}$

Gli elementi vuoti (nulli) nel primo argomento si spostano in fondo. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

►Sphere (Visualizza come vettore sferico)

Catalogo >

Vettore ►Sphere

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Sphere.

Visualizza il vettore di riga o colonna nel formato sferico [$p \angle\theta \angle\phi$].

[1 2 3] ►Sphere
[3.74166 \angle 1.10715 \angle 0.640522]

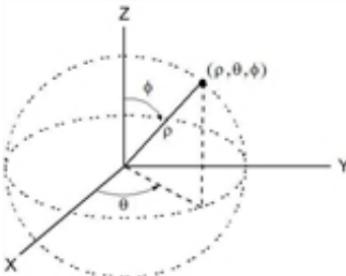
►Sphere (Visualizza come vettore sferico)

Catalogo >

Vettore deve avere dimensione 3 e può essere un vettore di riga o colonna.

Nota: ►**Sphere** è un'istruzione nel formato di visualizzazione, ma non è una funzione di conversione. Tale istruzione può essere impiegata solo alla fine di una riga di introduzione.

$\left(\begin{matrix} 2 & \angle \frac{\pi}{4} & 3 \\ 3.60555 & \angle 0.785398 & \angle 0.588003 \end{matrix} \right) \rightarrow \text{Sphere}$



sqrt() (Radice quadrata)

Catalogo >

sqrt(Valore1)⇒valore

$$\sqrt{4} \quad 2$$

sqrt(ListaI)⇒lista

$$\sqrt{\{9,2,4\}} = \{3,1.41421,2\}$$

Restituisce la radice quadrata dell'argomento.

In una lista, restituisce le radici quadrate di tutti gli elementi di *Listal*.

Nota: vedere anche **Modello di radice quadrata**, pagina 1.

stat.results (Risultati dell'analisi statistica)

Catalogo >

stat.results

Visualizza risultati di un calcolo statistico.

I risultati sono visualizzati sotto forma di coppie nome-valore. I nomi specifici mostrati dipendono dall'ultima funzione o comando statistico applicato.

È possibile copiare un nome o un valore e incollarlo in altre posizioni.

Nota: evitare di definire variabili con gli stessi nomi delle variabili utilizzate per l'analisi statistica. In alcuni casi, potrebbe prodursi un errore. I nomi di variabile utilizzati per l'analisi statistica sono elencati nella seguente tabella.

xlist:= {1,2,3,4,5} {1,2,3,4,5}

ylist:= {4,8,11,14,17} {4,8,11,14,17}

LinRegMx xlist, ylist, 1: stat.results

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r ² "	0.996109
"r"	0.998053
"Resid"	"{...}"

stat.values	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	"{-0.4,0.4,0.2,0.,-0.2}"

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBlock
stat.AdjR ²	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r ²	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSIInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.ox	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.oy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.ox1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.ox2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Sx	stat.X̄
stat.b9	stat.FBlock	stat.p̂	stat.Sx ²	stat.X̄1
stat.b10	stat.Fcol	stat.p̂1	stat.Sxy	stat.X̄2
stat.bList	stat.FInteract	stat.p̂2	stat.Sy	stat.X̄Diff
stat.χ ²	stat.FreqReg	stat.p̂Diff	stat.Sy ²	stat.X̄List
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg

stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat. \bar{y}
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat. \hat{y}
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat. \hat{y} List
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

Nota: ogni volta che Foglio elettronico calcola risultati statistici, l'applicazione copia le variabili di gruppo "stat." in un gruppo "stat#", dove # è un numero che viene incrementato automaticamente. Questa funzione consente di mantenere i risultati precedenti mentre si eseguono calcoli multipli.

stat.values (Valori dei risultati)

Catalogo >

stat.values

Per un esempio vedere stat.results.

Visualizza una matrice dei valori calcolati per l'ultima funzione o comando statistico calcolato.

Diversamente da **stat.results**, **stat.values** omette i nomi associati ai valori.

È possibile copiare un valore e incollarlo in altre posizioni.

stDevPop() (Deviazione standard della popolazione)

Catalogo >

stDevPop(Lista[, listaFreq])⇒espressione

In modalità angolo in radianti:

Restituisce la deviazione standard degli elementi di *Lista*.

stDevPop({1,2,5, 6,3,-2})	3.59398
stDevPop({1.3,2.5, -6.4},{3,2,5})	4.11107

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

Nota: *Lista* deve avere almeno due elementi. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

stDevPop() (Deviazione standard della popolazione)

Catalogo >

stDevPop(*MatriceI*[,
matriceFreq]) \Rightarrow *matrice*

Restituisce un vettore di riga delle deviazioni standard della popolazione delle colonne di *MatriceI*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *MatriceI*.

Nota: *MatriceI* deve avere almeno due righe. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

stDevPop $\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{pmatrix}$
[3.26599 2.94392 1.63299]

stDevPop $\begin{pmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{pmatrix} \begin{pmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{pmatrix}$
[2.52608 5.21506]

stDevSamp() (Deviazione standard del campione)

Catalogo >

stDevSamp(*Lista*[,
listaFreq]) \Rightarrow *espressione*

Restituisce la deviazione standard del campione di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

Nota: *Lista* deve avere almeno due elementi. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

stDevSamp(*MatriceI*[,
matriceFreq]) \Rightarrow *matrice*

Restituisce un vettore di riga delle deviazioni standard del campione delle colonne di *MatriceI*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *MatriceI*.

Nota: *MatriceI* deve avere almeno due righe. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

stDevSamp({1,2,5,-6,3,-2}) 3.937
stDevSamp({1.3,2.5,-6.4},{3,2,5})

4.33345

stDevSamp $\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{pmatrix}$
[4. 3.60555 2.]

stDevSamp $\begin{pmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{pmatrix} \begin{pmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{pmatrix}$
[2.7005 5.44695]

Stop**Catalogo >** **Stop**

Istruzione di programmazione: termina il programma il programma.

Stop non è ammesso nelle funzioni.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

<i>i:=0</i>	<i>Done</i>
Define <i>prog1()</i> =Prgm For <i>i</i> ,1,10,1 If <i>i</i> =5 Stop EndFor EndPrgm	<i>Done</i>
<i>prog1()</i>	<i>Done</i>

<i>i</i>	5
----------	---

Store (Memorizza)**Vedere → (memorizza), pagina 213.****string() (Da espressione a stringa)****Catalogo >** **string(*Espr*)⇒stringa**

Semplifica *Espr* e restituisce il risultato come una stringa di caratteri.

string(1.2345)	"1.2345"
string(1+2)	"3"

subMat() (Sottomatrice)**Catalogo >** **subMat[*Matrice1*[, *rigaInizio*] [, *colInizio*] [, *rigaFine*] [, *collFine*]] ⇒matrice**

Restituisce la sottomatrice specificata di *Matrice1*.

Impostazioni predefinite: *rigaInizio*=1, *colInizio*=1, *rigaFine*=ultima riga, *collFine*=ultima colonna.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
subMat(<i>m1</i> ,1,1,3,2)	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
subMat(<i>m1</i> ,2,2)	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)**Vedere Σ(), pagina 205.**

sum() (Somma)

Catalogo >

sum(Lista[, Inizio[, Fine]])⇒espressione

Restituisce la somma degli elementi di *Lista*.

Inizio e *Fine* sono opzionali e specificano un intervallo di elementi.

Un argomento vuoto produce un risultato vuoto. Gli elementi vuoti (nulli) di *Lista* vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

sum(MatriceI[, Inizio[, Fine]])⇒matrice

Restituisce un vettore di riga contenente le somme degli elementi nelle colonne della *MatriceI*.

Inizio e *Fine* sono opzionali e specificano un intervallo di elementi.

Un argomento vuoto produce un risultato vuoto. Gli elementi vuoti (nulli) di *MatriceI* vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

sum({1,2,3,4,5})	15
sum({a,2·a,3·a})	"Error: Variable is not defined"
sum(seq(n,n,1,10))	55
sum({1,3,5,7,9},3)	21

sum({1 2 3 4 5 6})	[5 7 9]
sum({1 2 3 4 5 6 7 8 9})	[12 15 18]
sum({1 2 3 4 5 6 7 8 9},2,3)	[11 13 15]

sumIf()

Catalogo >

sumIf(Lista,Criteri[, listaSomma])⇒valore

Restituisce la somma cumulata di tutti gli elementi di *Lista* che soddisfano i *Criteri* specificati. Opzionalmente è possibile specificare una lista alternativa, *ListaSomma*, per fornire gli elementi da accumulare.

Lista può essere un'espressione, una lista o una matrice. *listaSomma*, se specificata, deve avere le stesse dimensioni di *Lista*.

Criteri può essere:

- Un valore, un'espressione o una stringa. Ad esempio, **34** accumula solo quegli elementi di *Lista* che sono semplificati nel valore 34.
- Un'espressione booleana contenente il simbolo **?** come segnaposto di ciascun

sumIf({1,2,e,3,π,4,5,6},2.5<?<4.5)	12.859874482
sumIf({1,2,3,4},2<?<5,{10,20,30,40})	70

elemento. Ad esempio, `?<10` accumula solo quegli elementi di *Lista* che sono minori di 10.

Quando un elemento di *Lista* soddisfa i *Criteri*, esso viene aggiunto alla somma accumulata. Se si include *listaSomma*, l'elemento corrispondente di *listaSomma* viene aggiunto invece alla somma.

In Foglio elettronico, è possibile utilizzare un intervallo di celle al posto di *Lista* e di *listaSomma*.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

Nota: vedere anche **countIf()**, pagina 32.

system(*Valore1* [, *Valore2* [, *Valore3* [, ...]]])

Restituisce un sistema di equazioni, formattato come una lista. È inoltre possibile creare un sistema utilizzando un modello.

T

Matrice1 **T** \Rightarrow *matrice*

Restituisce la trasposta dei complessi coniugati di *Matrice1*.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^{\top} \quad \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando `@t`.

tan() (Tangente)

Tasto 

tan(Valore1)⇒valore

tan(Lista1)⇒lista

tan(Valore1) restituisce la tangente dell'argomento.

tan(Lista1) restituisce una lista delle tangenti di tutti gli elementi di *Lista1*.

Nota: l'argomento è interpretato come angolo in gradi, gradianti o radianti, a seconda della modalità angolo correntemente impostata. Si può usare °, G o † per escludere tale impostazione provvisoriamente.

In modalità angolo in gradi:

$$\tan\left(\left(\frac{\pi}{4}\right)_r\right) \quad 1.$$

$$\tan(45) \quad 1.$$

$$\tan(\{0,60,90\}) \quad \{0.,1.73205,undef\}$$

In modalità angolo in gradi centesimali:

$$\tan\left(\left(\frac{\pi}{4}\right)_g\right) \quad 1.$$

$$\tan(50) \quad 1.$$

$$\tan(\{0,50,100\}) \quad \{0.,1.,undef\}$$

In modalità angolo in radianti:

$$\tan\left(\frac{\pi}{4}\right) \quad 1.$$

$$\tan(45^\circ) \quad 1.$$

$$\tan\left\{\pi, \frac{\pi}{3}, \frac{\pi}{2}, \frac{\pi}{4}\right\} \quad \{0.,1.73205,0.,1.\}$$

tan

(matriceQuadrata1)⇒matriceQuadrata

Restituisce la tangente della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare la tangente di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti:

$$\tan\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \quad \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

tan⁻¹⁽⁾ (Arcotangente)

Tasto 

tan⁻¹(Valore1)⇒valore

tan⁻¹(Lista1)⇒lista

In modalità angolo in gradi:

$$\tan^{-1}(1) \quad 45$$

tan⁻¹() (Arcotangente)

Tasto 

tan⁻¹(Valore1) restituisce l'angolo la cui tangente è *Valore1*.

tan⁻¹(Lista1) restituisce una lista dell'inversa della tangente di ciascun elemento di *Lista1*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradianti o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arctan(...)**.

tan⁻¹

(matriceQuadrata1)⇒*matriceQuadrata*

Restituisce la tangente inversa della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare la tangente inversa di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in gradianti (gradi centesimali):

tan⁻¹(1)

50

In modalità angolo in radianti:

tan⁻¹({0,0,2,0,5}) {0,0,197396,0,463648}

In modalità angolo in radianti:

tan⁻¹	$\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$
	$\begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$

tanh() (Tangente iperbolica)

Catalogo > 

tanh(Valore1)⇒*valore*

tanh(Lista1)⇒*lista*

tanh(Valore1) restituisce la tangente iperbolica dell'argomento.

tanh(Lista1) restituisce una lista delle tangenti iperboliche di ciascun elemento di *Lista1*.

tanh

(matriceQuadrata1)⇒*matriceQuadrata*

Restituisce la tangente iperbolica della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare la tangente iperbolica di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

In modalità angolo in radianti:

tanh	$\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$
	$\begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$

tanh() (Tangente iperbolica)

Catalogo >

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

tanh⁻¹() (Arcotangente iperbolica)

Catalogo >

tanh⁻¹(Valore1)⇒valore

tanh⁻¹(Lista1)⇒lista

tanh⁻¹(Valore1) restituisce l'inversa della tangente iperbolica dell'argomento.

tanh⁻¹(Lista1) restituisce una lista dell'inversa della tangente iperbolica di ciascun elemento di *Lista1*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arctanh(...)**.

tanh⁻¹
(matriceQuadrata1)⇒matriceQuadrata

Restituisce la tangente iperbolica inversa della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare la tangente iperbolica inversa di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità formato rettangolare complesso:

tanh⁻¹(0) 0.

tanh⁻¹({1,2,1,3})

{undef,0.518046-1.5708·i,0.346574-1.570

Per vedere l'intero risultato, premere ▲, quindi utilizzare ▲ e ▶ per spostare il cursore.

In modalità angolo in radianti e in modalità formato rettangolare complesso:

tanh⁻¹
$$\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$$

[-0.099353+0.164058·i 0.267834-1.4908
-0.087596-0.725533·i 0.479679-0.94730·i
0.511463-2.08316·i -0.878563+1.7901]

Per vedere l'intero risultato, premere ▲, quindi utilizzare ▲ e ▶ per spostare il cursore.

tCdf() (Probabilità di distribuzione t di Student)

Catalogo >

tCdf(estremoInf,estremoSup,gl)⇒numero
se *estremoInf* e *estremoSup* sono numeri,
lista se *estremoInf* e *estremoSup* sono liste

Calcola la probabilità della distribuzione *t* di Student tra il *estremoInf* e il *estremoSup* per i gradi di libertà *gl* specificati.

Per $P(X \leq valoreSuperiore)$, impostare *valoreInferiore* = -9E999.

Text

Catalogo >

Text*stringaPrompt[, FlagDispl]*

Istruzione di programmazione: Sospende il programma e visualizza la stringa di caratteri *stringaPrompt* in una finestra di dialogo.

Selezionando **OK**, l'esecuzione del programma continua.

L'argomento opzionale *flag* può essere un'espressione.

- Se *FlagDispl* viene omesso o dà come risultato **1**, il messaggio testuale viene aggiunto alla cronologia di Calcolatrice.
- Se *FlagDispl* dà come risultato **0**, il messaggio testuale non viene aggiunto alla cronologia.

Se il programma richiede di digitare una risposta, vedere **Request**, pagina 136 oppure **RequestStr**, pagina 138.

Nota: è possibile utilizzare questo comando all'interno di un programma definito dall'utente, ma non di una funzione.

Definire un programma che si arresta momentaneamente per visualizzare ciascuno dei cinque numeri casuali in una finestra di dialogo.

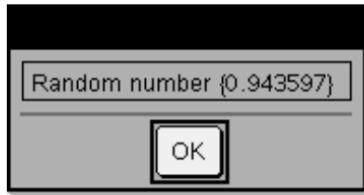
All'interno del modello **Prgm...EndPrgm**, completare ciascuna riga premendo  invece di **enter**. Sulla tastiera del computer, mantenere premuto **Alt** e premere **Invio**.

```
Define text_demo()=Prgm  
  For i,1,5  
    strinfo:="Random number " &  
    string(rand(i))  
    Text strinfo  
  EndFor  
EndPrgm
```

Eseguire il programma:

```
text_demo()
```

Esempio di finestra di dialogo:



Then

Vedere If, pagina 74.

tInterval (Intervallo di confidenza t)

Catalogo >

tInterval *Lista[,Freq[,livelloConfidenza]]*

tInterval (Intervallo di confidenza t)

Catalogo >

(Input lista dati)

tInterval $\bar{x}, sx, n[, livelloConfidenza]$

(Input statistiche riepilogo)

Calcola un intervallo di confidenza t . Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza per una media non nota di una popolazione
stat. \bar{x}	Media del campione della successione di dati dalla distribuzione casuale normale
stat.ME	Margine di errore
stat.df	Gradi di libertà
stat.sx	Deviazione standard del campione
stat.n	Lunghezza della successione di dati con media del campione

tInterval_2Samp (Intervallo di confidenza t su due campioni)

Catalogo >

tInterval_2Samp $List1, Lista2[, Freq1[, Freq2[, livelloConfidenza[, Aggregata]]]]$

(Input lista dati)

tInterval_2Samp $\bar{x}_1, sx1, n1, \bar{x}_2, sx2, n2[, livelloConfidenza[, Aggregata]]$

(Input statistiche riepilogo)

Calcola un intervallo di confidenza t su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Aggregata=1 aggrega le varianze;
Aggregata=0 non aggrega le varianze.

tInterval_2Samp (Intervallo di confidenza t su due campioni)

Catalogo >

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza contenente la probabilità di distribuzione del livello di confidenza
stat. \bar{x}_1 - \bar{x}_2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat.ME	Margine di errore
stat.df	Gradi di libertà
stat. \bar{x}_1 , stat. \bar{x}_2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat. σ_x1 , stat. σ_x2	Deviazioni standard dei campioni di <i>Lista 1</i> e <i>Lista 2</i>
stat.n1, stat.n2	Numero di campioni nelle successioni di dati
stat.sp	Deviazione standard aggregata. Calcola quando <i>Aggregata</i> = sì.

tPdf() (Densità di probabilità t di Student)

Catalogo >

tPdf(*ValX,gl*)⇒numero se *ValX* è un numero, lista se *ValX* è una lista

Calcola la funzione della densità di probabilità (pdf) per la distribuzione *t* di Student in corrispondenza di un valore *x* specificato con i gradi di libertà *gl* specificati.

trace() (Traccia)

Catalogo >

trace(*matriceQuadrata*)⇒valore

Restituisce la traccia (somma di tutti gli elementi sulla diagonale principale) di *matriceQuadrata*.

trace $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	15
<hr/>	
a:=12	12
<hr/>	
trace $\begin{bmatrix} a & 0 \\ 1 & a \end{bmatrix}$	24
<hr/>	

Try

blocco1

Else

blocco2

EndTry

Esegue *blocco1* a meno che non si verifichi un errore. In questo caso, l'esecuzione del programma viene trasferita a *blocco2*. La variabile di sistema *errCode* contiene il codice di errore che consente al programma di eseguire il ripristino dell'errore Per un elenco dei codici di errore, vedere "Codici di errore e messaggi", pagina 243.

blocco1 e *blocco2* possono essere o una singola istruzione o una serie di istruzioni separate dal carattere ":".

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Esempio 2

Per vedere i comandi **Try**, **ClrErr** e **PassErr** in funzione, il programma *eigenvals()* riportato sulla destra. Avviare il programma eseguendo ciascuna delle seguenti espressioni.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

Nota: vedere anche **ClrErr**, pagina 24 e **PassErr**, pagina 119.

Define *prog1()*=Prgm

Try

z:=z+1

Disp "z incremented."

Else

Disp "Sorry, z undefined."

EndTry

EndPrgm

Done

z:=1:prog1()

z incremented.

Done

DelVar *z:prog1()*

Sorry, z undefined.

Done

Define *eigenvals(a,b)=Prgm*

© Questo programma visualizza gli autovalori di A·B

Try

Disp "A= ",*a*

Disp "B= ",*b*

Disp " "

Disp "Eigenvalues of A·B are:",*eigVl(a*b)*

Else

If *errCode*=230 Then

Disp "Error: Product of A·B must be a square matrix"

ClrErr

```

Else
PassErr
EndIf
EndTry
EndPrgm

```

tTest (Verifica t)**tTest** $\mu0, Lista[, Freq[, Ipotesi]]$

(Input lista dati)

tTest $\mu0, \bar{x}, sx, n, [Ipotesi]$

(Input statistiche riepilogo)

Esegue una verifica dell'ipotesi su un'unica media μ non nota di una popolazione quando la deviazione standard σ della popolazione non è nota. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Viene verificata l'ipotesi nulla $H_0: \mu = \mu0$ in contrapposizione a una delle alternative seguenti:

Per $H_a: \mu < \mu0$, impostare *Ipotesi*<0Per $H_a: \mu \neq \mu0$ (default), impostare *Ipotesi*=0Per $H_a: \mu > \mu0$, impostare *Ipotesi*>0

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.t	$(\bar{x} - \mu0) / (\text{stdev} / \sqrt{n})$
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata

Variabile di output	Descrizione
stat.df	Gradi di libertà
stat. \bar{x}	Media del campione della sequenza di dati in <i>Lista</i>
stat.sx	Deviazione standard del campione della sequenza di dati
stat.n	Dimensione dei campioni

tTest_2Samp (Verifica t su due campioni)

Catalogo > 

tTest_2Samp *Lista1,Lista2[,Freq1[,Freq2[,Ipotesi[,Aggregata]]]]*

(Input lista dati)

tTest_2Samp $\bar{x}_1,sx_1,n_1,\bar{x}_2,sx_2,n_2[,Ipotesi[,Aggregata]]$

(Input statistiche riepilogo)

Esegue una verifica *t* su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Viene verificata l'ipotesi nulla $H_0: \mu_1 = \mu_2$ in contrapposizione a una delle alternative seguenti:

Per $H_a: \mu_1 < \mu_2$, impostare *Ipotesi*<0

Per $H_a: \mu_1 \neq \mu_2$ (default), impostare *Ipotesi*=0

Per $H_a: \mu_1 > \mu_2$, impostare *Ipotesi*>0

Aggregata=1 aggrega le varianze

Aggregata=0 non aggrega le varianze

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.t	Valore normale standard calcolato per la differenze delle medie

Variabile di output	Descrizione
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Gradi di libertà della statistica t
stat.Ȑx1, stat.Ȑx2	Medie dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</i>
stat.sx1, stat.sx2	Deviazioni standard dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</i>
stat.n1, stat.n2	Dimensione dei campioni
stat.sp	Deviazione standard aggregata. Calcolato quando <i>Aggregata</i> =1.

tvmFV()

Catalogo >

tvmFV(*N,I,PV,Pmt,[PpY],[CpY],[PmtAt]*)⇒*valore*

tvmFV(120,5,0,-500,12,12)

77641.1

Funzione finanziaria che calcola il valore futuro del denaro.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 176. Vedere anche **amortTbl()**, pagina 7.

tvmI()

Catalogo >

tvmI(*N,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*)⇒*valore*

tvmI(240,100000,-1000,0,0,12,12)

10.5241

Funzione finanziaria che calcola il tasso di interesse annuo.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 176. Vedere anche **amortTbl()**, pagina 7.

tvmN()

Catalogo >

tvmN(*I,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*)⇒*valore*

tvmN(5,0,-500,77641,12,12)

120.

Funzione finanziaria che calcola il numero di periodi di pagamento.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 176. Vedere anche **amortTbl()**, pagina 7.

tvmPmt($N, I, PV, FV, [PpY], [CpY], [PmtAt]$) \Rightarrow valore

tvmPmt(60,4,30000,0,12,12)

-552.496

Funzione finanziaria che calcola l'importo di ciascuna rata.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 176. Vedere anche **amortTbl()**, pagina 7.

tvmPV($N, I, Pmt, FV, [PpY], [CpY], [PmtAt]$) \Rightarrow valore

tvmPV(48,4,-500,30000,12,12)

-3426.7

Funzione finanziaria che calcola il valore presente.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 176. Vedere anche **amortTbl()**, pagina 7.

Argomento TVM *	Descrizione	Tipo di dati
N	Numero di periodi di pagamento	Numero reale
I	Tasso di interesse annuo	Numero reale
PV	Valore presente	Numero reale
Pmt	Importo della rata	Numero reale
FV	Valore futuro	Numero reale
PpY	Rate all'anno, impostazione predefinita=1	Numero intero > 0
CpY	Periodi di capitalizzazione all'anno, impostazione predefinita=1	Numero intero > 0

Argomento TVM *	Descrizione	Tipo di dati
<i>PmtAt</i>	Pagamento versato alla fine o all'inizio di ogni periodo di pagamento, impostazione predefinita=fine	Numero intero (0=fine, 1=inizio)

* Questi nomi di argomenti TVM sono simili ai nomi delle variabili TVM (come ad esempio **tvm.pv** e **tvm.pmt**) che sono utilizzate dal Risolutore finanziario dell'applicazione *Calculator*. Le funzioni finanziarie, tuttavia, non memorizzano automaticamente i valori degli argomenti o i risultati in variabili TVM.

TwoVar (Risultati a due variabili)

Catalogo > 

TwoVar *X, Y[, Freq [, Categoria, Includi]]*

Calcola le statistiche TwoVar. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici o di stringa di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Un elemento vuoto (nullo) in qualsiasi lista *X*, *Freq* o *Categoria* produce un corrispondente elemento vuoto in tutte queste liste. Un elemento vuoto (nullo) in qualsiasi lista da *X1* a *X20* produce un corrispondente elemento vuoto in tutte queste liste. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

Variabile di output	Descrizione
stat. \bar{X}	Media dei valori X
stat. x	Somma dei valori X
stat. x2	Somma dei valori x ²
stat.sx	Deviazione standard del campione di X
stat. x	Deviazione standard della popolazione di X
stat.n	Numero dei punti di dati
stat. \bar{y}	Media dei valori y
stat. y	Somma dei valori y
stat. y ²	Somma dei valori y ²
stat.sy	Deviazione standard del campione di y
stat. y	Deviazione standard della popolazione di y
stat. xy	Somma dei valori x · y
stat.r	Coefficiente di correlazione
stat.MinX	Minimo dei valori x
stat.Q ₁ X	1° quartile di x
stat.MedianX	Mediana di x
stat.Q ₃ X	3° quartile di x
stat.MaxX	Massimo dei valori x
stat.MinY	Minimo dei valori y
stat.Q ₁ Y	1° quartile di y
stat.MedY	Mediana di y
stat.Q ₃ Y	3° quartile di y
stat.MaxY	Massimo dei valori Y
stat. (x-) ²	Somma dei quadrati delle deviazioni dalla media di x
stat. (y-) ²	Somma dei quadrati delle deviazioni dalla media di y

U**unitV() (Vettore unità)****Catalogo > ****unitV(*VettoreI*)** \Rightarrow *vettore*Restituisce un vettore unità riga o colonna, a seconda della forma di *VettoreI*.*VettoreI* deve essere una matrice composta da una sola riga o da una sola colonna.

unitV([1 2 1])	[0.408248 0.816497 0.408248]
unitV([1 2 3])	[0.267261 0.534522 0.801784]

unLock**Catalogo > ****unLock***Var1 [, Var2] [, Var3] ...***unLock***Var.*

Sblocca le variabili o il gruppo di variabili specificate. Le variabili bloccate non possono essere modificate o eliminate.

Vedere **Lock**, pagina 93 e **getLockInfo()**, pagina 69.

<i>a:=65</i>	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a:=75</i>	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a:=75</i>	75
DelVar <i>a</i>	Done

V**varPop() (Varianza della popolazione)****Catalogo > ****varPop(*Lista*[, *listaFreq*])** \Rightarrow *espressione*

varPop({5,10,15,20,25,30}) 72.9167

Restituisce la varianza della popolazione di *Lista*.Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.**Nota:** *Lista* deve contenere almeno due elementi.

Se un elemento di una lista è vuoto (nullo), l'elemento viene ignorato, come pure l'elemento corrispondente nell'altra lista. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

varSamp() (Varianza campione)

Catalogo >

varSamp(*Lista*[, *listaFreq*]) \Rightarrow espressione

Restituisce la varianza campione di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

Nota: *Lista* deve contenere almeno due elementi.

Se un elemento di una lista è vuoto (nullo), l'elemento viene ignorato, come pure l'elemento corrispondente nell'altra lista. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

varSamp(*MatriceI*[, *matriceFreq*]) \Rightarrow matrice

Restituisce un vettore riga contenente la varianza campione di ciascuna colonna di *MatriceI*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *MatriceI*.

Nota: *MatriceI* deve contenere almeno due righe.

Se un elemento di una matrice è vuoto (nullo), l'elemento viene ignorato, come pure l'elemento corrispondente nell'altra matrice. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 233.

W

Wait

Catalogo >

Wait *tempoInSecondi*

Sospende l'esecuzione per un periodo di *tempoInSecondi* secondi.

Wait è particolarmente utile in un programma che necessita di un breve ritardo per consentire che i dati richiesti diventino disponibili.

varSamp({1,2,5,-6,3,-2})	31
	2
varSamp({1,3,5},{4,6,2})	68
	33

varSamp($\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}$)	[4.75 1.03 4]
varSamp($\begin{array}{c cc} -1.1 & 2.2 & [6 \quad 3] \\ 3.4 & 5.1 & [2 \quad 4] \\ -2.3 & 4.3 & [5 \quad 1] \end{array}$)	[3.91731 2.08411]

Wait

Catalogo >

L'argomento *tempoInSecondi* deve essere un'espressione che viene semplificata a un valore decimale da 0 a 100. Il comando arrotonda questo valore al decimo di secondo più vicino.

Per annullare un **Wait** in corso,

- **Palmare:** Tenere premuto il tasto [on] e premere [enter] più volte.
- **Windows®:** Tenere premuto il tasto **F12** e premere **Invio** più volte.
- **Macintosh®:** Tenere premuto il tasto **F5** e premere **Invio** più volte.
- **iPad®:** L'app mostra un prompt. È possibile continuare ad attendere o annullare.

Nota: È possibile utilizzare il comando **Wait** all'interno di un programma definito dall'utente, ma non di una funzione.

warnCodes ()

Catalogo >

warnCodes(*Espr1*, *varStato*)⇒*espressione*

Calcola l'espressione *Espr1*, restituisce il risultato e memorizza i codici di tutte le avvertenze generate nella variabile di lista *varStato*. Se non sono generate avvertenze, questa funzione assegna a *varStato* una lista vuota.

Espr1 può essere qualsiasi espressione matematica valida di TI-Nspire™ o TI-Nspire™ CAS. Non è possibile utilizzare un comando o una assegnazione come *Espr1*.

varStato deve essere un nome di variabile valido.

Per un elenco dei codici di avvertenza e dei messaggi associati, vedere pagina 252.

when() (Quando)

Catalogo >

when(*Condizione*, *risultatoVero* [, *risultatoFalso*][, *risultatoSconosciuto*])

seccount:=1.3

Wait seccount

Questo esempio fa accendere un LED verde per 0,5 secondi e quindi lo spegne.

Send "SET GREEN 1 ON"

Wait 0.5

Send "SET GREEN 1 OFF"

	warnCodes(det([1.23456E-999]),warn)
	1.23456E-999

warn { 10029 }

\Rightarrow espressione

Restituisce *risultatoVero*, *risultatoFalso* o *risultatoSconosciuto*, a seconda che *Condizione* sia vera, falsa o sconosciuta. Restituisce la voce inserita se gli argomenti immessi sono insufficienti per specificare un risultato appropriato.

Omettere sia *risultatoFalso* che *risultatoSconosciuto* se si desidera che l'espressione venga definita solamente nella regione in cui *Condizione* è vera.

Utilizzare **undef** come *risultatoFalso* per definire un'espressione che esegua una rappresentazione grafica solo in un intervallo.

when() è utile per la definizione di funzioni ricorsive.

when($x < 0, x + 3$) $x = 5$	undef
---------------------------------	-------

when($n > 0, n \cdot factorial(n - 1), 1$) $\rightarrow factorial(n)$	Done
---	------

factorial(3)	6
--------------	---

While

While *Condizione*

Blocco

EndWhile

Esegue le istruzioni di *Blocco* fino a quando *Condizione* è vera.

Blocco può essere una singola espressione o una serie di espressioni separate dal carattere ":".

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define sum_of_recip(n)=Func Local i,tempsum $1 \rightarrow i$ $0 \rightarrow tempsum$ While $i \leq n$ $tempsum + \frac{1}{i} \rightarrow tempsum$ $i+1 \rightarrow i$ EndWhile Return tempsum EndFunc	Done
---	------

sum_of_recip(3)	11
-----------------	----

xor

BooleanExpr1 xor BooleanExpr2 restituisce
Boolean expression

BooleanList1 xor BooleanList2 restituisce
Boolean list

BooleanMatrix1 xor BooleanMatrix2
restituisce *Boolean matrix*

Restituisce true se *Espressione booleana1* è vera e *Espressione booleana2* è falsa, o viceversa.

Restituisce false se gli argomenti sono entrambi veri o falsi. Restituisce un'espressione booleana semplificata se nessuno degli argomenti può essere risolto in vero o falso.

Nota: vedere **or**, pagina 117.

Intero1 xor Intero2 \Rightarrow *intero*

Confronta due interi reali bit per bit tramite un'operazione **xor**. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati bit corrispondenti, il risultato sarà 1 se uno dei due bit (ma non entrambi) è uguale a 1; se entrambi i bit sono uguali a 0 o a 1, il risultato sarà 0. Il valore restituito rappresenta il risultato dei bit e viene visualizzato nel modo Base che è stato impostato.

È possibile inserire gli interi in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli interi vengono considerati decimali (base 10).

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Per ulteriori informazioni, vedere ►**Base2**, pagina 17.

true xor true	false
5>3 xor 3>5	true

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

In modalità base Bin:

0b100101 xor 0b100	0b100001
--------------------	----------

Nota: un numero binario può contenere fino a 64 cifre (oltre al prefisso 0b). Un numero esadecimale può contenere fino ad 16 cifre.

Nota: vedere **or**, pagina 117.

Z**zInterval (Intervallo di confidenza Z)**Catalogo > 

zInterval σ ,*Lista*[,*Freq*[,*livelloConfidenza*]]

(Input lista dati)

zInterval σ, \bar{x}, n [,*livelloConfidenza*]

(Input statistiche riepilogo)

Calcola un intervallo di confidenza z . Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160)

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
<i>stat.CLower</i> , <i>stat.CUpper</i>	Intervallo di confidenza per una media non nota di una popolazione
<i>stat.</i>	Media del campione della successione di dati dalla distribuzione casuale normale
<i>stat.ME</i>	Margine di errore
<i>stat.sx</i>	Deviazione standard del campione
<i>stat.n</i>	Lunghezza della successione di dati con media del campione
<i>stat.σ</i>	Deviazione standard nota della popolazione per la sequenza di dati <i>Lista</i>

zInterval_1Prop (Intervallo di confidenza z per una proporzione)Catalogo > 

zInterval_1Prop x, n [,*livelloConfidenza*]

Calcola un intervallo di confidenza z per una proporzione. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

x è un intero non negativo.

zInterval_1Prop (Intervallo di confidenza z per una proporzione)Catalogo > 

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza contenente la probabilità di distribuzione del livello di confidenza
stat. \hat{p}	Proporzione calcolata di casi favorevoli
stat.ME	Margine di errore
stat.n	Numero di campioni nelle successioni di dati

zInterval_2Prop (Intervallo di confidenza z per due proporzioni)Catalogo > 

zInterval_2Prop $x1, n1, x2, n2$
[,livelloConfidenza]

Calcola l'intervallo di confidenza z per due proporzioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

$x1$ e $x2$ sono interi non negativi.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza contenente la probabilità di distribuzione del livello di confidenza
stat. \hat{p} Diff	Differenza calcolata tra le proporzioni
stat.ME	Margine di errore
stat. $\hat{p}1$	Stima della proporzione del primo campione
stat. $\hat{p}2$	Stima della proporzione del secondo campione
stat.n1	Dimensione del campione nella successione di dati uno
stat.n2	Dimensione del campione nella successione di dati due

**zInterval_2Samp (Intervallo di
confidenza z su due campioni)**

Catalogo >

zInterval_2Samp $\sigma_1, \sigma_2, List1, List2[, Freq1, Freq2, [livelloConfidenza]]$

(Input lista dati)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}_1, n1, \bar{x}_2, n2, [livelloConfidenza]$

(Input statistiche riepilogo)

Calcola l'intervallo di confidenza z su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza contenente la probabilità di distribuzione del livello di confidenza
stat. \bar{x}_1 - \bar{x}_2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat.ME	Margine di errore
stat. \bar{x}_1 , stat. \bar{x}_2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat. σx_1 , stat. σx_2	Deviazioni standard dei campioni di <i>List1</i> e <i>List2</i>
stat.n1, stat.n2	Numero di campioni nelle successioni di dati
stat.r1, stat.r2	Deviazioni standard note della popolazione per la successione di dati <i>List1</i> e <i>List2</i>

zTest (Verifica z)

Catalogo >

zTest $\mu_0, \sigma, Lista, [Freq, Ipotesi]$

(Input lista dati)

zTest $\mu_0, \sigma, \bar{x}, n, [Ipotesi]$

(Input statistiche riepilogo)

zTest (Verifica z)

Catalogo > 

Esegue una verifica z con frequenza *listaFreq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Viene verificata l'ipotesi nulla $H_0: \mu = \mu_0$ in contrapposizione a una delle alternative seguenti:

Per $H_a: \mu < \mu_0$, impostare *Ipotesi<0*

Per $H_a: \mu \neq \mu_0$ (default), impostare *Ipotesi=0*

Per $H_a: \mu > \mu_0$, impostare *Ipotesi>0*

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$
stat.P Valore	Probabilità minima in cui l'ipotesi nulla può essere rifiutata
stat. \bar{x}	Media del campione della sequenza di dati in <i>Lista</i>
stat.sx	Deviazione standard del campione della sequenza di dati. Restituita solo per input di <i>Dati</i> .
stat.n	Dimensione dei campioni

zTest_1Prop (Verifica z per una proporzione)

Catalogo > 

zTest_1Prop *p0,x,n,[Ipotesi]*

Esegue una verifica z per una proporzione. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

x è un intero non negativo.

Viene verificata l'ipotesi nulla $H_0: p = p_0$ in contrapposizione a una delle alternative seguenti:

Per $H_a: p > p_0$, impostare *Ipotesi>0*

zTest_1Prop (Verifica z per una proporzione)

Catalogo > 

Per $H_a: p \neq p0$ (default), impostare
Ipotesi=0

Per $H_a: p < p0$, impostare *Ipotesi<0*

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.p0	Proporzione della popolazione ipotizzata
stat.z	Valore normale standard calcolato per la proporzione
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat. \hat{p}	Proporzione attesa del campione
stat.n	Dimensione dei campioni

zTest_2Prop (Verifica z per due proporzioni)

Catalogo > 

zTest_2Prop x1,n1,x2,n2[,Ipotesi]

Esegue una verifica z per due proporzioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

$x1$ e $x2$ sono interi non negativi.

Viene verificata l'ipotesi nulla $H_0: p1 = p2$ in contrapposizione a una delle alternative seguenti:

Per $H_a: p1 > p2$, impostare *Ipotesi>0*

Per $H_a: p1 \neq p2$ (default), impostare
Ipotesi=0

Per $H_a: p < p0$, impostare *Ipotesi<0*

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.z	Valore normale standard calcolato per la differenza delle proporzioni
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat. \hat{p}_1	Stima della proporzione del primo campione
stat. \hat{p}_2	Stima della proporzione del secondo campione
stat. \hat{p}	Stima delle proporzioni del campione aggregate
stat.n1, stat.n2	Numero di campioni presi nelle prove 1 e 2

zTest_2Samp (Verifica su due campioni)

Catalogo > 

zTest_2Samp $\sigma_1, \sigma_2, Lista1, Lista2[, Freq1, Freq2[, Ipotesi]]$

(Input lista dati)

zTest_2Samp $\sigma_1, \sigma_2, \bar{x}_1, n1, \bar{x}_2, n2[, Ipotesi]$

(Input statistiche riepilogo)

Esegue una verifica z su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 160).

Viene verificata l'ipotesi nulla $H_0: \mu_1 = \mu_2$ in contrapposizione a una delle alternative seguenti:

Per $H_a: \mu_1 < \mu_2$, impostare *Ipotesi*<0

Per $H_a: \mu_1 \neq \mu_2$ (default), impostare *Ipotesi*=0

Per $H_a: \mu_1 > \mu_2$, *Ipotesi*>0

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 233.

Variabile di output	Descrizione
stat.z	Valore normale standard calcolato per la differenze delle medie

Variabile di output	Descrizione
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat. \bar{x}_1 , stat. \bar{x}_2	Medie dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</i>
stat.sx1, stat.sx2	Deviazioni standard dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</i>
stat.n1, stat.n2	Dimensione dei campioni

Simboli

+ (addizione)	Tasto
<i>Valore1 + Valore2⇒valore</i>	56
Restituisce la somma dei due argomenti.	56+4
	60
	60+4
	64
	64+4
	68
	68+4
	72
<i>Listal + Lista2⇒lista</i>	$\left\{ 22, \pi, \frac{\pi}{2} \right\} \rightarrow l1$
	{22,3.14159,1.5708}
<i>Matrice1 + Matrice2⇒matrice</i>	$\left\{ 10,5, \frac{\pi}{2} \right\} \rightarrow l2$
Restituisce una lista (o matrice) contenente le somme degli elementi corrispondenti di <i>Listal</i> e <i>Lista2</i> (oppure di <i>Matrice1</i> e <i>Matrice2</i>).	{10,5,1.5708}
	$l1+l2$
	{32,8.14159,3.14159}
<i>Gli argomenti devono avere uguali dimensioni.</i>	
<i>Valore + Listal⇒lista</i>	15+{10,15,20}
	{25,30,35}
<i>Listal + Valore⇒lista</i>	{10,15,20}+15
	{25,30,35}
<i>Restituisce una lista contenente le somme di <i>Valore</i> e di ciascun elemento di <i>Listal</i>.</i>	
<i>Valore + Matrice1⇒matrice</i>	$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
<i>Matrice1 + Valore⇒matrice</i>	
Restituisce una matrice nella quale <i>Valore</i> viene sommato a ciascun elemento sulla diagonale di <i>Matrice1</i> . <i>Matrice1</i> deve essere quadrata.	
Nota: utilizzare .+ (punto più) per aggiungere un'espressione a ciascun elemento.	
- (sottrazione)	Tasto
<i>Valore1 - Valore2⇒valore</i>	6-2
Restituisce <i>Valore1</i> meno <i>Valore2</i> .	$\pi - \frac{\pi}{6}$
	2.61799

-(sottrazione)Tasto **-***Lista1 - Lista2⇒lista**Matrice1 - Matrice2⇒matrice*

Sottrae ciascun elemento di *Lista2* (o di *Matrice2*) dal corrispondente elemento di *Lista1* (o di *Matrice1*) e restituisce il risultato.

Gli argomenti devono avere uguali dimensioni.

*Valore - Lista1⇒lista**Lista1 - Valore⇒lista*

Sottrae ciascun elemento di *Lista1* da *Valore* oppure sottrae *Valore* da ciascun elemento di *Lista1*, quindi restituisce una lista contenente i risultati.

*Valore - Matrice1⇒matrice**Matrice1 - Valore⇒matrice*

Valore - Matrice1 restituisce una matrice di numero volte *Valore* la matrice di identità, meno *Matrice1*. *Matrice1* deve essere quadrata.

Matrice1 - Valore restituisce una matrice di *Valore* volte la matrice d'identità, sottratta da *Matrice1*. *Matrice1* deve essere quadrata.

Nota: Utilizzare **.** (punto meno) per sottrarre un'espressione da ciascun elemento.

$$\begin{array}{l} \left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10,5, \frac{\pi}{2} \right\} \\ [3 \ 4] - [1 \ 2] \end{array} \quad \left\{ 12, -1.85841, 0. \right\} \quad [2 \ 2]$$

$$\begin{array}{l} 15 - \{ 10, 15, 20 \} \\ \{ 10, 15, 20 \} - 15 \end{array} \quad \left\{ 5, 0, -5 \right\} \quad \left\{ -5, 0, 5 \right\}$$

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

·(moltiplicazione)Tasto **x***Valore1 · Valore2⇒valore*

Restituisce il prodotto di due argomenti.

Lista1 · Lista2⇒lista

Restituisce una lista contenente i prodotti degli elementi corrispondenti di *Lista1* e *Lista2*.

Le liste devono avere uguali dimensioni.

$$2 \cdot 3.45 \quad 6.9$$

$$\{ 1., 2, 3 \} \cdot \{ 4, 5, 6 \} \quad \{ 4, 10, 18 \}$$

·(moltiplicazione)**Tasto** *Matrice1 · Matrice2⇒matrice*Restituisce il prodotto di matrice di *Matrice1* per *Matrice2*.

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 7 & 8 \\ 7 & 8 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 42 & 48 \\ 105 & 120 \end{bmatrix}$$

Il numero di colonne di *Matrice1* deve essere uguale al numero di righe di *Matrice2*.*Valore · Listal⇒lista*

$$\pi \cdot \{4, 5, 6\} = \{12.5664, 15.708, 18.8496\}$$

*Listal · Valore⇒lista*Restituisce una lista contenente i prodotti di *Valore* per ciascun elemento di *Listal*.*Valore · Matrice1⇒matrice**Matrice1 · Valore⇒matrice*Restituisce una matrice contenente i prodotti di *Valore* per ciascun elemento di *Matrice1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 = \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$6\text{-identity}(3) = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

Nota: Utilizzare . (punto moltiplicazione) per moltiplicare un'espressione per ciascun elemento.**/ (divisione)****Tasto** *Valore1 / Valore2⇒valore*

$$\frac{2}{3.45} = 0.57971$$

Restituisce il quoziente di *Valore1* diviso per *Valore2*.**Nota:** vedere anche **Modello di frazione**, pagina 1.*Listal / Lista2⇒lista*

$$\frac{\{1,2,3\}}{\{4,5,6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

Restituisce una lista contenente i quozienti di *Listal* diviso per *Lista2*.

Le liste devono avere uguali dimensioni.

Valore / Listal ⇒ lista

$$\frac{6}{\{3,6,\sqrt{6}\}} = \{2,1,2.44949\}$$

Listal / Valore ⇒ lista

$$\frac{\{7,9,2\}}{7 \cdot 9 \cdot 2} = \left\{\frac{1}{18}, \frac{1}{14}, \frac{1}{63}\right\}$$

Restituisce una lista contenente i quozienti di *Valore* diviso per *Listal* o quelli di *Listal* diviso per *Valore*.

/ (divisione)**Tasto** *Valore / Matrice1 \Rightarrow matrice*

$$\begin{bmatrix} 7 & 9 & 2 \\ \end{bmatrix}$$

$7 \cdot 9 \cdot 2$

$$\begin{bmatrix} 1 & 1 & 1 \\ \hline 18 & 14 & 63 \\ \end{bmatrix}$$

Matrice1 / Valore \Rightarrow matrice

Restituisce una matrice contenente i quozienti di *Matrice1* per *Valore*.

Nota: utilizzare `. /` (punto divisione) per dividere un'espressione per ciascun elemento.

 \wedge (elevamento a potenza)**Tasto** *Valore1 \wedge Valore2 \Rightarrow valore*

$$4^2$$

$\{2,4,6\}^{\{1,2,3\}}$

16

$$\{2,16,216\}$$

Restituisce il primo argomento elevato alla potenza del secondo argomento.

Nota: vedere anche **Modello di esponente**, pagina 1.

In una lista, restituisce gli elementi di *Listal* elevati alla potenza dei corrispondenti elementi di *List2*.

Nel dominio reale, le potenze frazionarie con esponenti ridotti e denominatori dispari utilizzano l'ambito reale rispetto a quello principale, impiegata nel modo complesso.

Valore \wedge Listal \Rightarrow lista

$$\pi^{\{1,2,-3\}}$$

$$\{3.14159, 9.8696, 0.032252\}$$

Restituisce *Valore* elevato alla potenza degli elementi di *Listal*.

Listal \wedge Valore \Rightarrow lista

$$\{1,2,3,4\}^{-2}$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

Restituisce gli elementi di *Listal* elevati alla potenza di *Valore*.

\wedge (elevamento a potenza)

matriceQuadrata1 \wedge *intero* \Rightarrow *matrice*

Restituisce *matriceQuadrata1* elevata alla potenza di *intero*.

matriceQuadrata1 deve essere una matrice quadrata.

Se *intero* = -1, viene calcolata l'inversa della matrice.

Se *intero* < -1, viene calcolata l'inversa della matrice a una potenza positiva appropriata.

Tasto

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2$	$\begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1}$	$\begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2}$	$\begin{bmatrix} \frac{11}{4} & -\frac{5}{4} \\ 2 & 2 \\ -\frac{15}{4} & \frac{7}{4} \end{bmatrix}$

x² (quadrato)

Valore1 $^2 \Rightarrow$ *valore*

Restituisce il quadrato dell'argomento.

Listal 2 \Rightarrow *lista*

Restituisce una lista contenente i quadrati degli elementi di *Listal*.

matriceQuadrata1 $^2 \Rightarrow$ *matrice*

Restituisce la matrice *matriceQuadrata1* elevata al quadrato. Ciò non equivale a calcolare il quadrato di ogni elemento. Utilizzare $.^2$ per calcolare il quadrato di ogni elemento.

Tasto

4^2	16
$\{2,4,6\}^2$	{4,16,36}
$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$	$\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$
$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} .^2$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$

.+ (punto addizione)

Matrice1 .+ *Matrice2* \Rightarrow *matrice*

Valore .+ *Matrice1* \Rightarrow *matrice*

Matrice1 .+ *Matrice2* restituisce una matrice che è la somma di ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

Tasti

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix}$	$\begin{bmatrix} 11 & 32 \\ 23 & 44 \end{bmatrix}$
$5 .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix}$	$\begin{bmatrix} 15 & 35 \\ 25 & 45 \end{bmatrix}$

Valore .+ *Matrice1* restituisce una matrice che è la somma di *Valore* e di ciascun elemento di *Matrice1*.

. - (punto sottrazione).

Tasti  

Matrice1 . - Matrice2 \Rightarrow matrice

Valore . - Matrice1 \Rightarrow matrice

Matrice1 . - Matrice2 restituisce una matrice che è la differenza tra ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

Valore . - Matrice1 restituisce una matrice che è la differenza di *Valore* e di ciascun elemento di *Matrice1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} . - \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} -9 & -18 \\ -27 & -36 \end{bmatrix}$$

$$5 . - \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} -5 & -15 \\ -25 & -35 \end{bmatrix}$$

. · (punto moltiplicazione)

Tasti  

Matrice1 . · Matrice2 \Rightarrow matrice

Valore . · Matrice1 \Rightarrow matrice

Matrice1 . · Matrice2 restituisce una matrice che è il prodotto di ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

Valore . · Matrice1 restituisce una matrice contenente i prodotti di *Valore* per ciascun elemento di *Matrice1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} . \cdot \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 10 & 40 \\ 90 & 160 \end{bmatrix}$$

$$5 . \cdot \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 50 & 100 \\ 150 & 200 \end{bmatrix}$$

. / (punto divisione)

Tasti  

Matrice1 . / Matrice2 \Rightarrow matrice

Valore . / Matrice1 \Rightarrow matrice

Matrice1 . / Matrice2 restituisce una matrice che è il quoziente di ciascuna coppia di elementi corrispondenti di *Matrice1* e di *Matrice2*.

Valore . / Matrice1 restituisce una matrice che è il quoziente di *Valore* diviso per ciascun elemento di *Matrice1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} . / \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} \end{bmatrix}$$

$$5 . / \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{8} \end{bmatrix}$$

$\{\{1,10,100\}\}\%$

{0.01,0.1,1.}

= (uguale)

$Espr1 = Espr2 \Rightarrow$ espressione booleana

$List1 = Lista2 \Rightarrow$ lista booleana

$Matrice1 = Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere uguale a $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere uguale ad $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Tasto

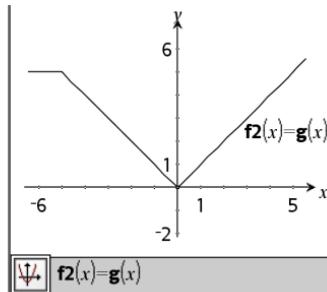
Esempio di funzione che usa gli operatori relazionali: $=, \neq, <, \leq, >, \geq$

Define $g(x) = \text{Func}$

```
If x ≤ -5 Then
  Return 5
ElseIf x > -5 and x < 0 Then
  Return -x
ElseIf x ≥ 0 and x ≠ 10 Then
  Return x
ElseIf x = 10 Then
  Return 3
EndIf
EndFunc
```

Done

Risultato della rappresentazione grafica $g(x)$



$f_2(x) = g(x)$

≠ (diverso)

$Espr1 \neq Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di “=” (uguale).

$List1 \neq Lista2 \Rightarrow$ lista booleana

$Matrice1 \neq Matrice2 \Rightarrow$ Matrice booleana

Restituisce true se $Espr1$ è determinata in modo da non essere uguale a $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da essere uguale a $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando /=

< (minore di)

$Espr1 < Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di “=” (uguale).

$Listal < Lista2 \Rightarrow$ lista booleana

$Matrice1 < Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere minore di $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere maggiore o uguale a $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

≤ (minore di o uguale a)

$Espr1 \leq Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di “=” (uguale).

$Listal \leq Lista2 \Rightarrow$ lista booleana

$Matrice1 \leq Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere minore o uguale a $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere maggiore di $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

\leq (minore di o uguale a)Tasti  

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando $<=$

 $>$ (maggiore di)Tasti  

$Espr1 > Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di " $=$ " (uguale).

$List1 > Lista2 \Rightarrow$ lista booleana

$Matrice1 > Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere maggiore di $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere minore o uguale a $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

 \geq (maggiore di o uguale a)Tasti  

$Espr1 \geq Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di " $=$ " (uguale).

$List1 \geq Lista2 \Rightarrow$ lista booleana

$Matrice1 \geq Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere maggiore o uguale a $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere minore o uguale a $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando $>=$

\Rightarrow (implicazione logica)

ctrl = tasti

$BooleanExpr1 \Rightarrow BooleanExpr2$ restituisce Boolean expression

$BooleanList1 \Rightarrow BooleanList2$ restituisce Boolean list

$BooleanMatrix1 \Rightarrow BooleanMatrix2$ restituisce Boolean matrix

$Integer1 \Rightarrow Integer2$ restituisce Integer

5>3 or 3>5	true
5>3 \Rightarrow 3>5	false
3 or 4	7
3 \Rightarrow 4	-4
{1,2,3} or {3,2,1}	{3,2,3}
{1,2,3} \Rightarrow {3,2,1}	{-1,-1,-3}

Valuta l'espressione **not** <argomento1> **or** <argomento2> e restituisce true, false oppure una forma semplificata dell'equazione.

Nel caso di liste e matrici, restituisce un confronto elemento per elemento.

Nota: È possibile inserire questo operatore dalla tastiera digitando =>

\Leftrightarrow (doppia implicazione logica, XNOR)

ctrl = tasti

$BooleanExpr1 \Leftrightarrow BooleanExpr2$ restituisce Boolean expression

$BooleanList1 \Leftrightarrow BooleanList2$ restituisce Boolean list

$BooleanMatrix1 \Leftrightarrow BooleanMatrix2$ restituisce Boolean matrix

$Integer1 \Leftrightarrow Integer2$ restituisce Integer

5>3 xor 3>5	true
5>3 \Leftrightarrow 3>5	false
3 xor 4	7
3 \Leftrightarrow 4	-8
{1,2,3} xor {3,2,1}	{2,0,2}
{1,2,3} \Leftrightarrow {3,2,1}	{-3,-1,-3}

Restituisce la negazione di un'operazione **XOR** logica riguardo i due argomenti.

Restituisce true, false oppure una forma semplificata dell'equazione.

Nel caso di liste e matrici, restituisce un confronto elemento per elemento.

Nota: È possibile inserire questo operatore dalla tastiera digitando <=>

! (fattoriale)**Tasto** $Valore1! \Rightarrow valore$

$5!$	120
$\{\{5,4,3\}\}!$	$\{120,24,6\}$
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}!$	$\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

 $Listal! \Rightarrow lista$ $Matrice1! \Rightarrow matrice$

Restituisce il fattoriale dell'argomento.

In una lista o in una matrice, restituisce rispettivamente una lista o una matrice dei fattoriali di ciascun elemento.

& (aggiunge)**Tasti** $Stringa1 \& Stringa2 \Rightarrow stringa$

"Hello "&"Nick"

"Hello Nick"

Restituisce una stringa di testo, aggiungendo *Stringa2* a *Stringa1*.**d() (derivata)****Catalogo** > $d(Espr1, Var[, Ordine]) \mid$
 $Var=Valore \Rightarrow valore$

$$\frac{d}{dx}(|x|)|_{x=0} \quad \text{undef}$$

 $d(Espr1, Var[, Ordine]) \Rightarrow valore$

$$x:=0: \frac{d}{dx}(|x|) \quad \text{undef}$$

 $d(Lista1, Var[, Ordine]) \Rightarrow lista$

$$x:=3: \frac{d}{dx}(\{x^2, x^3, x^4\}) \quad \{6, 27, 108\}$$

 $d(Matrice1, Var[, Ordine]) \Rightarrow matrice$ Tranne quando si utilizza la prima sintassi, è necessario memorizzare un valore numerico nella variabile *Var* prima di calcolare *d()*. Vedere gli esempi.*d()* può essere utilizzato per calcolare numericamente la derivata di primo e di secondo ordine in un punto con metodi di differenziazione automatica.*Ordine*, se incluso, deve essere = **1** o **2**. L'impostazione predefinita è **1**.**Nota:** è possibile inserire questa funzione dalla tastiera del computer digitando **derivative (...)**.**Nota:** Vedere anche **Derivata prima, pagina 5** o **Derivata seconda, pagina 6**.

d() (derivata)

Catalogo >

Nota: l'algoritmo **d()** ha una limitazione: funziona in modo ricorsivo tramite l'espressione non semplificata, calcolando il valore numerico della derivata prima (e seconda, se applicabile) e di ciascuna sottoespressione, il che può produrre un risultato imprevisto.

Si consideri l'esempio sulla destra. La derivata prima di $x \cdot (x^2+x)^{1/3}$ con $x=0$ è uguale a 0. Tuttavia, poiché la derivata prima della sottoespressione $(x^2+x)^{1/3}$ è indefinita con $x=0$, e questo valore è utilizzato per calcolare la derivata dell'espressione totale, **d()** riporta il risultato come indefinito e visualizza un messaggio di avvertenza.

Se si incontra questa limitazione, verificare la soluzione graficamente. È anche possibile provare a utilizzare **centralDiff()**.

$$\frac{d}{dx} \left(x \cdot (x^2+x)^{\frac{1}{3}} \right) \Big|_{x=0} \quad \text{undef}$$

$$\text{centralDiff}\left(x \cdot (x^2+x)^{\frac{1}{3}}, x\right) \Big|_{x=0} \quad 0.000033$$

j() (integrale)

Catalogo >

j(Esp1, Var, Inferiore, Superiore) ⇒ valore

Restituisce l'integrale di *Esp1* rispetto alla variabile *Var* da *Inferiore* a *Superiore*. Può essere utilizzato per calcolare numericamente l'integrale definito, con lo stesso metodo di **nInt()**.

$$\int_0^1 x^2 \, dx \quad 0.333333$$

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **integral(...)**.

Nota: vedere anche **nInt()**, pagina 110 e **Modello di integrale definito**, pagina 6.

√() (radice quadrata)

Tasti

$\sqrt{Valore1} \Rightarrow \text{valore}$

$$\sqrt{4} \quad 2$$

$\sqrt{List1} \Rightarrow \text{lista}$

$$\sqrt{\{9,2,4\}} \quad \{3,1.41421,2\}$$

Restituisce la radice quadrata dell'argomento.

In una lista, restituisce le radici quadrate di tutti gli elementi di *Listal*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **sqrt** (...)

Nota: vedere anche **Modello di radice quadrata**, pagina 1.

 $\prod()$ (prodSeq)

$\prod(Espr1, Var, Basso, Alto) \Rightarrow$ espressione

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **prodSeq** (...).

Calcola *Espr1* per ciascun valore di *Var* da *Basso* ad *Alto* e restituisce il prodotto dei risultati.

Nota: vedere anche **Modello di prodotto** (\prod), pagina 5.

$\prod(Espr1, Var, Basso, Basso-1) \Rightarrow 1$

$\prod(Espr1, Var, Basso, Alto) \Rightarrow 1 / \prod(Espr1, Var, Alto+1, Basso-1)$ if *Alto* < *Basso-1*

Le formule di prodotto utilizzate sono derivate dal seguente riferimento:

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

Catalogo >

$$\prod_{n=1}^{5} \left(\frac{1}{n} \right) \quad \frac{1}{120}$$

$$\prod_{n=1}^{5} \left(\left\{ \frac{1}{n}, n, 2 \right\} \right) \quad \left\{ \frac{1}{120}, 120, 32 \right\}$$

$$\prod_{k=4}^{3} (k) \quad 1$$

$$\prod_{k=4}^{1} \left(\frac{1}{k} \right) \quad 6$$

$$\prod_{k=4}^{1} \left(\frac{1}{k} \right) \cdot \prod_{k=2}^{4} \left(\frac{1}{k} \right) \quad \frac{1}{4}$$

$\Sigma()$ (sumSeq)

Catalogo >

$\Sigma(Expr1, Var, Bass, Alto) \Rightarrow$ espressione

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **sumSeq(...)**.

Calcola *Expr1* per ciascun valore di *Var* da *Bass* ad *Alto* e restituisce la somma dei risultati.

Nota: vedere anche **Modello di somma**, pagina 5.

$\Sigma(Expr1, Var, Bass, Bass-1) \Rightarrow 0$

$\Sigma(Expr1, Var, Bass, Alto) \Rightarrow -\Sigma(Expr1, Var, Alto+1, Bass-1)$ if *Alto* < *Bass-1*

$$\sum_{n=1}^5 \left(\frac{1}{n} \right)$$

137
60

$$\sum_{k=4}^3 (k)$$

0

$$\sum_{k=4}^1 (k)$$

-5

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k)$$

4

Le formule di sommatoria utilizzate sono derivate dal seguente riferimento:

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$\SigmaInt()$

Catalogo >

$\SigmaInt(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [valoreArrotondato]) \Rightarrow$ valore

$\SigmaInt(1,3,12,4.75,20000,,12,12)$ -213.48

ΣInt

(

NPmt1

,*NPmt2,tabellaAmmortamento*) \Rightarrow valore

Funzione di ammortamento che calcola la somma dell'interesse pagato durante un intervallo specifico di pagamenti.

NPmt1 e *NPmt2* definiscono l'inizio e la fine dell'intervallo dei pagamenti.

N, I, PV, Pmt, FV, PpY, CpY e PmtAt sono descritti nella tabella degli argomenti TVM, pagina 176.

$\Sigma\text{Int}()$

Catalogo >

- Se si omette Pmt , viene utilizzata l'impostazione predefinita $Pmt=\text{tvmPmt}$ ($N,I,PV,FV,PpY,CpY,PmtAt$).
- Se si omette FV , viene utilizzata l'impostazione predefinita $FV=0$.
- Le impostazioni predefinite di PpY , CpY e $PmtAt$ sono le stesse delle funzioni TVM.

valoreArrotondato specifica il numero di cifre decimali di arrotondamento.

Impostazione predefinita=2.

ΣInt

($NPmt1, NPmt2, tabellaAmmortamento$)

calcola la somma degli interessi sulla base della tabella di ammortamento

tabellaAmmortamento. L'argomento

tabellaAmmortamento deve essere una

matrice avente la forma descritta in

amortTbl(), pagina 7.

Nota: vedere anche $\Sigma\text{Prn}()$, sotto, e $\text{Bal}()$, pagina 16.

$tbl:=\text{amortTbl}(12,12,4.75,20000,,12,12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\text{Int}(1,3,tbl)$ -213.48

$\Sigma\text{Prn}()$

Catalogo >

$\Sigma\text{Prn}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [valoreArrotondato]) \Rightarrow valore$

$\Sigma\text{Prn}(1,3,12,4.75,20000,,12,12)$ -4916.28

ΣPrn

(

NPmt1

,*NPmt2,tabellaAmmortamento*) $\Rightarrow valore$

Funzione di ammortamento che calcola la somma del capitale versato durante un intervallo specifico di pagamenti.

NPmt1 e *NPmt2* definiscono l'inizio e la fine dell'intervento dei pagamenti.

N, I, PV, Pmt, FV, PpY, CpY e PmtAt sono descritti nella tabella degli argomenti TVM, pagina 176.

- Se si omette Pmt , viene utilizzata l'impostazione predefinita $Pmt=\text{tvmPmt}$ ($N,I,PV,FV,PpY,CpY,PmtAt$).

$tbl:=\text{amortTbl}(12,12,4.75,20000,,12,12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\text{Prn}(1,3,tbl)$ -4916.28

- Se si omette *FV*, viene utilizzata l'impostazione predefinita *FV=0*.
- Le impostazioni predefinite di *PpY*, *CpY* e *PmtAt* sono le stesse delle funzioni TVM.

valoreArrotondato specifica il numero di cifre decimali di arrotondamento.
Impostazione predefinita=2.

ΣPrn

(*NPmt1,NPmt2,tabellaAmmortamento*)
calcola la somma del capitale versato sulla base della tabella di ammortamento *tabellaAmmortamento*. L'argomento *tabellaAmmortamento* deve essere una matrice avente la forma descritta in **amortTbl()**, pagina 7.

Nota: vedere anche **ΣInt()**, sopra, e **Bal()**, pagina 16.

(conversione indiretta)

*stringaNomeVariabile*

Questo operatore, che si riferisce alla variabile chiamata *stringaNomeVariabile*, permette di creare e di modificare le variabili dall'interno di una funzione mediante le stringhe.

Tasti

<i>xyz:=12</i>	12
<i>#("x" & "y" & "z")</i>	12

Crea o fa riferimento alla variabile xyz.

<i>10→r</i>	10
<i>"r" →s1</i>	"r"
<i>#s1</i>	10

Restituisce il valore della variabile (r) il cui nome è memorizzato nella variabile s1.

E (notazione scientifica)

mantissaEesponente

Permette di inserire un numero in notazione scientifica. Il numero viene interpretato come *mantissa* × 10^{esponente}.

Tasto

23000.	23000.
2300000000.+4.1E15	4.1E15
3·10 ⁴	30000

E (notazione scientifica)**Tasto**

Suggerimento: se si desidera inserire una potenza di 10 senza ottenere un valore decimale per risultato, utilizzare 10^{interno} .

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @E. Per esempio, digitare 2.3@E4 per inserire 2.3E4.

g (gradianti)**Tasto** *Espr1g* \Rightarrow espressione

In modalità di misurazione degli angoli in gradi, gradianti o radianti:

Espr1g \Rightarrow espressione

$$\cos(50^\circ) \quad 0.707107$$

Listal1g \Rightarrow lista

$$\cos(\{0,100^\circ,200^\circ\}) \quad \{1,0.,-1.\}$$

Matrice1g \Rightarrow matrice

Questa funzione consente di utilizzare un angolo espresso in gradianti pure essendo attiva la modalità di misurazione degli angoli in gradi o in radianti.

In modalità angolo in radianti, moltiplica *Espr1* per $\pi/200$.

In modalità angolo in gradi, moltiplica *Espr1* per $g/100$.

In modalità gradianti, restituisce *Espr1* senza modifiche.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando @g.

r (radianti)**Tasto** *Valore1r* \Rightarrow valore

In modalità angolo in gradi, gradianti o radianti:

Listal1r \Rightarrow lista

$$\cos\left(\frac{\pi}{4^r}\right) \quad 0.707107$$

Matrice1r \Rightarrow matrice

$$\cos\left(\left\{0^r, \left(\frac{\pi}{12}\right)^r, -(\pi)^r\right\}\right) \quad \{1,0.965926,-1.\}$$

'(radianti)

Tasto **[1]**

Questa funzione consente di utilizzare un angolo espresso in radianti pur essendo attiva la modalità di misurazione degli angoli in gradi o in gradi.

In modalità angolo in gradi, moltiplica l'argomento per $180/\pi$.

In modalità angolo in radianti, restituisce l'argomento immutato.

In modalità angolo in gradi, moltiplica l'argomento per $200/\pi$.

Suggerimento: ricorrere al simbolo ' se si desidera forzare l'uso dei radianti in una definizione, indipendentemente dalla modalità prevalente nella funzione.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando **@r**.

° (gradi)

Tasto **[1]**

Valore $I^\circ \Rightarrow$ *valore*

Lista $I^\circ \Rightarrow$ *lista*

Matrice $I^\circ \Rightarrow$ *matrice*

Questa funzione consente di utilizzare un angolo espresso in gradi pur essendo attiva la modalità di misurazione degli angoli in gradi o in radianti.

In modalità angolo in radianti, moltiplica l'argomento per $\pi/180$.

In modalità angolo in gradi, restituisce l'argomento immutato.

In modalità angolo in gradi, moltiplica l'argomento per $10/9$.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando **@d**.

In modalità angolo in gradi, gradi o radianti:

$$\cos(45^\circ) \quad 0.707107$$

In modalità angolo in radianti:

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) \\ \{1., 0.707107, 0., 0.864976\}$$

$^{\circ}, ', ''$ (gradi/primi/secondi)

Tasti

$gg^{\circ}pp'ss.ss'' \Rightarrow$ espressione

In modalità angolo in gradi:

gg un numero positivo o negativo

25 $^{\circ}$ 13'17.5"

25.2215

pp un numero non negativo

25 $^{\circ}$ 30'

51

2

ss.ss un numero non negativo

Restituisce $gg + (pp/60) + (ss.ss/3600)$.

Questo formato di introduzione in base -60 consente di:

- Inserire un angolo in gradi/primi/secondi indipendentemente dalla modalità di misurazione degli angoli corrente.
- Inserire un orario nella forma ore/minuti/secondi.

Nota: far seguire ss.ss da due apostrofi ('') e non dal simbolo di virgolette ("").

\angle (angolo)

Tasti

[Raggio, \angle Angolo] \Rightarrow vettore

In modalità angolo in radianti e con il formato vettore impostata su:

(formato polare)

rettangolare

[Raggio, \angle Angolo, Z_Coordinate] \Rightarrow vettore

[5 \angle 60 $^{\circ}$ \angle 45 $^{\circ}$]

[1.76777 3.06186 3.53553]

(formato cilindrico)

[Raggio, \angle Angolo, \angle Angolo] \Rightarrow vettore

cilindrico

(formato sferico)

[5 \angle 60 $^{\circ}$ \angle 45 $^{\circ}$]

[3.53553 \angle 1.0472 3.53553]

Restituisce le coordinate nella forma di un vettore a seconda dell'impostazione della modalità formato vettore: rettangolare, cilindrico o sferico.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando @<.

sferico

[5 \angle 60 $^{\circ}$ \angle 45 $^{\circ}$]

[5. \angle 1.0472 \angle 0.785398]

(Grandezza \angle Angolo) \Rightarrow valoreComplesso
(formato polare)

In modalità angolo in radianti e in modalità formato rettangolare complesso:

∠ (angolo)

Tasti

ctrl



Inserisce un valore complesso in formato polare ($r\angle\theta$). *Angolo* viene interpretato conformemente alla modalità di misurazione degli angoli impostata.

$$5+3 \cdot i \cdot \left(10 \angle \frac{\pi}{4}\right) \quad -2.07107 - 4.07107 \cdot i$$

_ (trattino basso come elemento vuoto)

Vedere “Elementi vuoti (nulli)”, pagina 233.

_ (trattino basso come designatore di unità)

Tasti

ctrl

**Variabile_**

Se *Variabile* non ha un valore, essa verrà trattata come se indicasse un numero complesso. Per impostazione predefinita, senza **_**, la variabile verrà considerata un numero reale.

Assumendo che *z* non sia definito:

real(<i>z</i>)	<i>z</i>
real(<i>z</i> _)	real(<i>z</i> _)
imag(<i>z</i>)	0
imag(<i>z</i> _)	imag(<i>z</i> _)

Se a *Variabile* è associato un valore, il carattere **_** viene ignorato e *Variabile* conserverà il tipo di dati originale.

10^()

Catalogo >**10^(Valore1)⇒valore**10^{1.5}

31.6228

10^(Listal)⇒lista

Restituisce 10 elevato alla potenza dell'argomento.

In una lista, restituisce 10 elevato alla potenza degli elementi di *Listal*.

10^(matriceQuadrata1)⇒matriceQuadrata

Restituisce 10 elevato alla potenza di *matriceQuadrata1*. Ciò non equivale a calcolare 10 elevato alla potenza di ciascun elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

$$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1.14336\text{e}7 & 8.17155\text{e}6 & 6.67589\text{e}6 \\ 9.95651\text{e}6 & 7.11587\text{e}6 & 5.81342\text{e}6 \\ 7.65298\text{e}6 & 5.46952\text{e}6 & 4.46845\text{e}6 \end{bmatrix}$$

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

\wedge^{-1} (reciproco)

Catalogo >

Valore1 $\wedge^{-1} \Rightarrow$ *valore* $\begin{pmatrix} 3.1 \end{pmatrix}^{-1}$

0.322581

Listal $\wedge^{-1} \Rightarrow$ *lista*

Restituisce il reciproco dell'argomento.

In una lista, restituisce i reciproci degli elementi di *Listal*.*matriceQuadrata1* $\wedge^{-1} \Rightarrow$ *matriceQuadrata*Restituisce l'inversa di *matriceQuadrata1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$$

matriceQuadrata1 deve essere una matrice quadrata non singolare.**| (operatore di vincolo)** **tasti***Expr* | *BooleanExpr1* $x+1|x=3$ 4[**and***BooleanExpr2*]... $x+55|x=\sin(55)$ 54.0002*Expr* | *BooleanExpr1* [**or***BooleanExpr2*]...

Il simbolo di vincolo (“|”) serve da operatore binario. L'operando a sinistra di | è un'espressione. L'operando a destra di | definisce una o più relazioni destinate ad avere un effetto sulla semplificazione dell'espressione. Le relazioni multiple dopo | devono essere collegate da operatori logici “**and**” o “**or**”.

L'operatore di vincolo fornisce tre tipi primari di funzionalità:

- Sostituzioni
- Vincoli d'intervallo
- Esclusioni

Le sostituzioni si presentano nella forma di equazione, come per esempio $x=3$ o $y=\sin(x)$. I migliori risultati si ottengono quando la parte sinistra è una variabile semplice.

Expr | *Variabile* = *valore* sostituirà *valore* per ogni ricorrenza di *Variabile* in *Expr*.

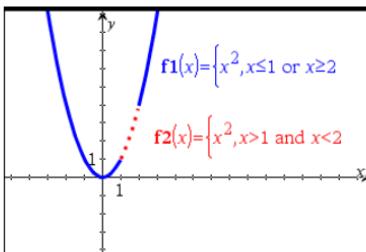
 $x^3 - 2 \cdot x + 7 \rightarrow f(x)$ Done $f(x)|x=\sqrt{3}$ 8.73205

I (operatore di vincolo)

ctrl tasti

I vincoli d'intervallo si presentano nella forma di una o più disequazioni, collegate da operatori logici "and" o "or". I vincoli d'intervallo permettono inoltre la semplificazione, la quale sarebbe altrimenti non valida o non calcolabile.

$$\begin{array}{ll} \text{nSolve}\left(x^3+2 \cdot x^2-15 \cdot x=0, x\right) & 0 . \\ \text{nSolve}\left(x^3+2 \cdot x^2-15 \cdot x=0, x\right)|_{x>0 \text{ and } x<5} & 3 . \end{array}$$



Le esclusioni utilizzano l'operatore relazionale "diverso da" (\neq or \neq) per escludere dalla valutazione un valore specifico.

→ (memorizza)

Tasti ctrl var

Valore → *Var*

$$\frac{\pi}{4} \rightarrow myvar \quad 0.785398$$

List →

$$2 \cdot \cos(x) \rightarrow y1(x) \quad \text{Done}$$

Matrix → *Var*

$$\{1,2,3,4\} \rightarrow lst5 \quad \{1,2,3,4\}$$

Espr → *Funzione(Param1,...)*

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow matg \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Lista → *Funzione(Param1,...)*

$$\text{"Hello"} \rightarrow str1 \quad \text{"Hello"}$$

Matrice → *Funzione(Param1,...)*

Se la variabile *Var* non esiste, *Var* viene creata e inizializzata in *Valore*, *Lista* o *Matrice*.

Se la variabile *Var* esiste già e non è bloccata o protetta, l'operatore ne sostituisce il contenuto con *Valore*, *Lista* o *Matrice*.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando `=:` come scelta rapida. Per esempio, digitare `pi/4 =: Miavar`.

:= (assegna)Tasti *Var := Valore*myvar:= $\frac{\pi}{4}$.785398*Var := Lista*

yI(x):=2·cos(x) Done

Var := Matrice

lst5:={1,2,3,4} {1,2,3,4}

*Funzione(Param1,...) := Espr*matg:=[1 2 3] [1 2 3]
[4 5 6] [4 5 6]*Funzione(Param1,...) := Lista*

str1:="Hello" "Hello"

Funzione(Param1,...) := Matrice

Se la variabile *Var* non esiste, *Var* viene creata e inizializzata in *Valore*, *Lista* o *Matrice*.

Se la variabile *Var* esiste già e non è bloccata o protetta, l'operatore ne sostituisce il contenuto con *Valore*, *Lista* o *Matrice*.

© (commento)Tasti **© [testo]**

© considera *testo* come una riga di commento che consente di annotare le funzioni e i programmi creati.

© può trovarsi all'inizio o in qualsiasi altro punto della riga. Tutto quanto si trova a destra del segno © fino alla fine della riga viene considerato come commento.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define g(n)=Func

© Declare variables
Local i,result
result:=0
For i,1,n,1 ©Loop n times
result:=result+i^2
EndFor
Return result
EndFunc

Done

g(3)

14

0b, 0hTasti , tasti **0b numeroBinario**

In modalità base Dec:

0h numeroEsadecimale

0b10+0hF+10

27

In modalità base Bin:

0b, 0h

Indica, rispettivamente, un numero binario o un numero esadecimale. Per inserire un numero binario o esadecimale, inserire il prefisso 0b o 0h indipendentemente dalla modalità Base che è stata impostata. Senza prefisso, un numero viene considerato decimale (base 10).

I risultati vengono visualizzati nella modalità Base che è stata impostata.

Tasti **0** **B**, tasti **0** **H**

0b10+0hF+10

0b11011

In modalità base Esadecimale:

0b10+0hF+10

0h1B

TI-Nspire™ CX II - Comandi di Disegna

Si tratta di un documento integrativo della Guida di riferimento di TI-Nspire™ e della Guida di riferimento di TI-Nspire™ CAS. Tutti i comandi di TI-Nspire™ CX II verranno integrati e pubblicati nella versione 5.1 della Guida di riferimento di TI-Nspire™ e della Guida di riferimento di TI-Nspire™ CAS.

Programmazione grafica

Ai palmari TI-Nspire™ CX II e alle applicazioni da desktop TI-Nspire™ sono stati aggiunti nuovi comandi per la programmazione grafica.

I palmari TI-Nspire™ CX II passeranno a questa modalità grafico durante l'esecuzione dei comandi grafici e ritorneranno alla modalità di esecuzione del programma precedente una volta completato il programma.

Durante l'esecuzione del programma, sulla barra superiore verrà visualizzato il messaggio "In esecuzione..." Al completamento del programma, verrà mostrato il messaggio "Finito". La modalità grafico viene disattivata con la pressione di qualsiasi tasto.

- Il passaggio alla modalità grafico viene attivato automaticamente quando il programma TI-Basic individua uno dei comandi (grafico) di Disegna durante l'esecuzione.
- Questo passaggio avviene solo quando si esegue un programma dalla Calcolatrice; in un documento o una Calcolatrice nel Blocco note.
- Il passaggio alla modalità grafico avviene al termine del programma.
- La modalità grafico è disponibile solo nella Vista palmare di TI-Nspire™ CX II e nella Vista palmare di TI-Nspire™ CX II per desktop. Ciò significa che non è disponibile nella vista documento del computer o in PublishView (.tnsp) sul desktop o su iOS.
 - Se durante l'esecuzione di un programma TI-Basic nella modalità errata viene individuato un comando grafico, verrà visualizzato un messaggio di errore e il programma TI-Basic verrà chiuso.

Schermata grafica

La schermata grafica presenta un'intestazione nella parte superiore dello schermo che non può essere modificata dai comandi grafici.

L'area di disegno della schermata grafica viene cancellata (colore = 255.255.255) quando si inizializza la schermata grafica.

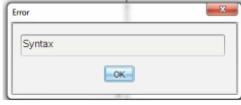
Schermata grafica	Predefinito
Altezza	212
Larghezza	318
Colore	bianco: 255.255.255

Impostazioni e vista predefinite

- Le icone di stato nella barra superiore (stato della batteria, stato Premere per Test, indicatore di rete, ecc.) non sono visibili durante l'esecuzione di un programma grafico.
- Colore predefinito per il disegno: nero (0,0,0)
- Stile della penna predefinito: normale, uniforme
 - Spessore: 1 (sottile), 2 (normale), 3 (molto spesso)
 - Stile 1 (uniforme), 2 (punteggiato), 3 (tratteggiato)
- Tutti i comandi di disegno utilizzano le impostazioni del colore e della penna correnti, vale a dire i valori predefiniti oppure i valori impostati mediante i comandi di TI-Basic.
- Il carattere del testo è fisso e non può essere modificato.
- Qualsiasi disegno eseguito nella schermata grafica viene tracciato all'interno di una finestra di ritaglio che presenta le stesse dimensioni dell'area di disegno della schermata grafica. Qualsiasi disegno che fuoriesce da quest'area di disegno ritagliata della schermata grafica non verrà tracciato. Non verrà visualizzato alcun messaggio di errore.
- Tutte le coordinate x,y specificate per i comandi di disegno vengono definite in modo che 0,0 si trovi nell'angolo superiore sinistro dell'area di disegno della schermata grafica.
 - **Eccezioni:**
 - **DrawText** utilizza le coordinate nell'angolo inferiore sinistro del riquadro di delimitazione per il testo.
 - **SetWindow** utilizza l'angolo inferiore sinistro della schermata.
- È possibile fornire tutti i parametri dei comandi come espressioni che restituiscono un numero che viene arrotondato al numero intero più vicino.

Messaggi di errore della schermata grafica

Se la convalida non riesce, viene visualizzato un messaggio di errore.

Messaggio di errore	Descrizione	Visualizza
Errore Sintassi	Se vengono rilevati eventuali errori di sintassi, l'applicazione di controllo visualizza un messaggio di errore e tenta di portare il cursore in prossimità del primo errore in modo che sia possibile correggerlo.	
Errore Argomenti mancanti	Nella funzione o nel comando mancano uno o più argomenti	Error Too few arguments The function or command is missing one or more arguments. OK
Errore Troppi argomenti	La funzione o il comando contiene un numero eccessivo di argomenti e non è possibile eseguirne il calcolo.	Error Too many arguments The function or command contains an excessive number of arguments and cannot be evaluated. OK
Errore Tipo di dati non valido	Un argomento è del tipo di dati errato.	Error Invalid data type An argument is of the wrong data type. OK

Comandi non validi in modalità grafico

Alcuni comandi vengono disattivati quando il programma passa alla modalità grafico. Se questi comandi vengono individuati durante la modalità grafico, verrà visualizzato un messaggio di errore e il programma verrà chiuso.

Comando non consentito	Messaggio di errore
Request	Impossibile eseguire Request in modalità grafico
RequestStr	Impossibile eseguire RequestStr in modalità grafico
Testo	Impossibile eseguire Text in modalità grafico

I comandi che eseguono la stampa del testo sulla Calcolatrice, **disp** e **dispAt**, saranno comandi supportati nella modalità grafico. Il testo immesso con questi comandi viene inviato alla schermata della Calcolatrice (non alla schermata grafica) e risulta visibile una volta che si chiude il programma e il sistema ritorna all'app Calcolatrice.

Cancellare**Catalogo >** 
CXII**Cancella $x, y, \text{larghezza}, \text{altezza}$**

Cancella l'intera schermata se non viene specificato alcun parametro.

Se vengono specificati $x, y, \text{larghezza}$ e altezza , verrà cancellato il rettangolo definito dai parametri.

Cancellare

Cancella l'intera schermata

Cancella $10,10,100,50$

Cancella l'area di un rettangolo con l'angolo in alto a sinistra su (10, 10) e con larghezza 100, altezza 50

DrawArc

Catalogo > CXII

DrawArc *x, y, larghezza, altezza, startAngle, arcAngle*

Disegna un arco all'interno del rettangolo di delimitazione definito con gli angoli di inizio e dell'arco forniti.

x, y: coordinata superiore sinistra del rettangolo di selezione

larghezza, altezza: dimensioni del rettangolo di delimitazione

L'"angolo dell'arco" definisce l'estensione dell'arco.

Questi parametri possono essere forniti come espressioni che restituiscono un numero che viene in seguito arrotondato al numero intero più vicino.

DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180



Vedere anche: [FillArc](#)

DrawCircle

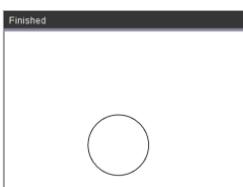
Catalogo > CXII

DrawCircle *x, y, raggio*

x, y: coordinata del centro

raggio: raggio della circonferenza

DrawCircle 150,150,40



Vedere anche: [FillCircle](#)

DrawLine

Catalogo > CXII

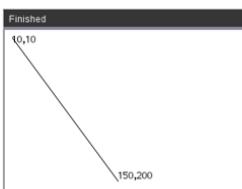
DrawLine $x1, y1, x2, y2$

Disegna una linea da $x1, y1, x2, y2$.

Espressioni che restituiscono un numero che viene in seguito arrotondato al numero intero più vicino.

Limits della schermata: se in base alle coordinate specificate qualsiasi parte della linea viene tracciata all'esterno della schermata grafica, tale parte verrà tagliata e non verrà visualizzato un messaggio di errore.

DrawLine $10,10,150,200$



DrawPoly

Catalogo > CXII

I comandi presentano due varianti:

DrawPoly $xlist, ylist$

oppure

DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$

Nota: la forma **DrawPoly** $xlist, ylist$ collegherà $x1, y1$ a $x2, y2$, $x2, y2$ a $x3, y3$ e così via.

Nota: **DrawPoly** $x1, y1, x2, y2, x3, y3...xn, yn$ **NON** verrà automaticamente collegata a $x1, y1$.

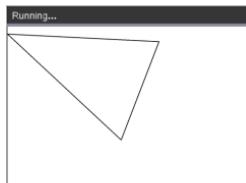
Espressioni che restituiscono un elenco di valori mobili reali
 $xlist, ylist$

Espressioni che restituiscono un unico valore mobile reale
 $x1, y1...xn, yn$ = coordinate per i vertici di un poligono

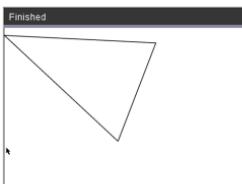
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

DrawPoly $xlist,ylist$



DrawPoly $0,10,200,20,150,150,0,10$



Nota: **DrawPoly**: dimensioni dell'input (larghezza/altezza) relative alle linee disegnate.

Le linee vengono tracciate in un riquadro di delimitazione attorno alla coordinata e alle dimensioni specificate in modo che le dimensioni effettive del poligono disegnato siano superiori a quelle della larghezza e dell'altezza.

Vedere anche: [FillPoly](#)

DrawRect

DrawRect *x, y, larghezza, altezza*

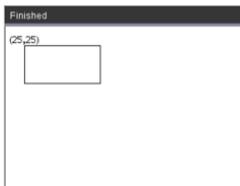
x, y: coordinata superiore sinistra del rettangolo

larghezza, altezza: larghezza e altezza del rettangolo (rettangolo tracciato verso il basso e a destra rispetto alla coordinata iniziale).

Nota: le linee vengono tracciate in un riquadro di delimitazione attorno alla coordinata e alle dimensioni specificate in modo che le dimensioni effettive del rettangolo disegnato siano superiori a quelle indicate dalla larghezza e dall'altezza.

Vedere anche: [FillRect](#)

DrawRect 25,25,100,50



DrawText

DrawText *x, y, exprOrString1*
[*exprOrString2*...]

x, y: coordinata dell'output del testo

Disegna il testo in *exprOrString* in corrispondenza delle coordinate *x, y* specificate.

DrawText 50,50,"Hello World"



Le regole per *exprOrString* sono identiche a quelle per **Disp** - **DrawText** può contenere più argomenti.

FillArc

Catalogo > CXII

FillArc *x, y, larghezza, altezza startAngle, arcAngle*

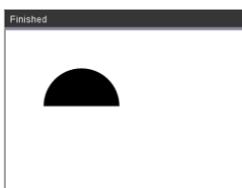
x, y: coordinata superiore sinistra del rettangolo di selezione

Disegna e riempie un arco all'interno del rettangolo di delimitazione definito con gli angoli di inizio e dell'arco forniti.

Il colore di riempimento predefinito è il nero. È possibile impostare il colore di riempimento con il comando [SetColor](#).

L'“angolo dell'arco” definisce l'estensione dell'arco.

FillArc 50,50,100,100,0,180

**FillCircle**

Catalogo > CXII

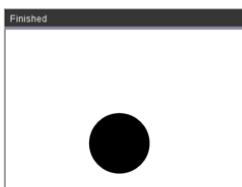
FillCircle *x, y, raggio*

x, y: coordinata del centro

Disegna e riempie una circonferenza in corrispondenza del centro specificato con il raggio specificato.

Il colore di riempimento predefinito è il nero. È possibile impostare il colore di riempimento con il comando [SetColor](#).

FillCircle 150,150,40



Here!

FillPoly

Catalogo > CXII

FillPoly *xlist, ylist*

oppure

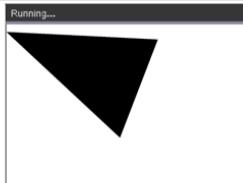
FillPoly *x1, y1, x2, y2, x3, y3...xn, yn*

Nota: utilizzare [SetColor](#) e [SetPen](#) per specificare la linea e il colore.

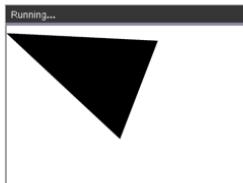
xlist:={0,200,150,0}

ylist:={10,20,150,10}

FillPoly xlist,ylist



`FillPoly 0,10,200,20,150,150,0,10`



FillRect

FillRect *x, y, larghezza, altezza*

x, y: coordinata superiore sinistra del rettangolo

larghezza, altezza: larghezza e altezza del rettangolo

Disegna e riempie un rettangolo il cui angolo superiore sinistro si trova in corrispondenza della coordinata specificata da (x,y)

Il colore di riempimento predefinito è il nero. È possibile impostare il colore di riempimento con il comando [SetColor](#).

Nota: utilizzare [SetColor](#) e [SetPen](#) per specificare la linea e il colore.

`FillRect 25,25,100,50`



getPlatform()**Catalogo >  CXII****getPlatform()**

getPlatform()

"dt"

Restituisce:

"dt" sulle applicazioni software da desktop

"hh" sui palmari TI-Nspire™ CX

"ios" sull'app TI-Nspire™ CX iPad®

PaintBuffer

Catalogo > 
CXII

PaintBuffer

Disegna il buffer grafico sulla schermata.

Questo comando viene utilizzato insieme a UseBuffer per aumentare la velocità di visualizzazione su schermo quando il programma genera più oggetti grafici.

UseBuffer

```
For n,1,10
x:=randInt(0,300)
y:=randInt(0,200)
raggio:=randInt(10,50)
```

```
Wait 0,5
```

```
DrawCircle x,y,raggio
```

```
EndFor
```

PaintBuffer

Questo programma visualizzerà tutti i 10 cerchi contemporaneamente.

Se il comando “UseBuffer” viene rimosso, ciascun cerchio sarà visualizzato come è disegnato.

Vedere anche: [UseBuffer](#)

PlotXY $x, y, forma$

x, y : coordinata per tracciare la forma

forma : un numero compreso tra 1 e 13 che specifica la forma

1 - Circonferenza piena

2 - Circonferenza vuota

3 - Quadrato pieno

4 - Quadrato vuoto

5 - Croce

6 - Più

7 - Sottile

8 - Punto medio, pieno

9 - Punto medio, vuoto

10 - Punto più grande, pieno

11 - Punto più grande, vuoto

12 - Il punto più grande, pieno

13 - Il punto più grande, vuoto

PlotXY 100,100,1

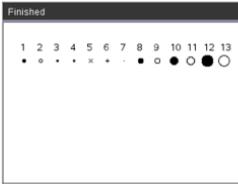


For n,1,13

DrawText 1+22*n,40,n

PlotXY 5+22*n,50,n

EndFor



SetColor

Catalogo > CXII

SetColor

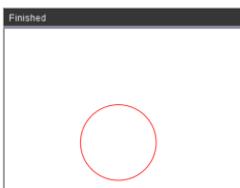
Valore rosso, verde, blu

I valori validi per il rosso, il verde e il blu sono compresi tra 0 e 255.

Imposta i colori per i comandi di Disegna successivi.

`SetColor 255,0,0`

`DrawCircle 150,150,100`

**SetPen**

Catalogo > CXII

SetPen

spessore, stile

spessore: $1 \leq$ spessore ≤ 3 | 1 è il più sottile, 3 è il più spesso

stile: 1 = Uniforme, 2 = Punteggiato, 3 = Tratteggiato

Imposta lo stile della penna per i comandi di Disegna successivi.

`SetPen 3,3`

`DrawCircle 150,150,50`

**SetWindow**

Catalogo > CXII

SetWindow

xMin, xMax, yMin, yMax

Definisce una finestra logica che esegue la mappatura all'area di disegno grafico. Tutti i parametri sono obbligatori.

Se una parte dell'oggetto disegnato si trova all'esterno della finestra, tale parte verrà tagliata (non mostrata) e non verrà visualizzato un messaggio di errore.

`SetWindow 0,160,0,120`

imposterà la finestra di output su 0,0 nell'angolo in basso a sinistra con larghezza di 160 e altezza di 120

`DrawLine 0,0,100,100`

`SetWindow 0,160,0,120`

`SetPen 3,3`

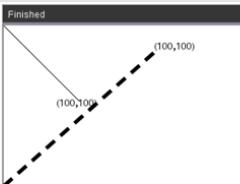
`DrawLine 0,0,100,100`

Se x_{Min} è maggiore o uguale a x_{Max} oppure y_{Min} è maggiore o uguale a y_{Max} , viene visualizzato un messaggio di errore.

Qualsiasi oggetto disegnato prima di un comando SetWindow non verrà tracciato nuovamente nella nuova configurazione.

Per ripristinare i valori predefiniti dei parametri della finestra, utilizzare:

SetWindow 0,0,0,0



UseBuffer

Catalogo > 
CXII

UseBuffer

Disegna in un buffer grafico fuori schermo invece che sullo schermo (per migliorare le prestazioni).

Questo comando viene utilizzato insieme a PaintBuffer per aumentare la velocità di visualizzazione su schermo quando il programma genera più oggetti grafici.

Con UseBuffer, tutti gli elementi grafici vengono visualizzati solo dopo aver eseguito il comando PaintBuffer successivo.

Il comando UseBuffer deve essere utilizzato una sola volta nel programma, vale a dire che ogni volta che si utilizza PaintBuffer non occorre utilizzare anche UseBuffer.

UseBuffer

```
For n,1,10
```

```
x:=randInt(0,300)
```

```
y:=randInt(0,200)
```

```
raggio:=randInt(10,50)
```

```
Wait 0,5
```

```
DrawCircle x,y,raggio
```

```
EndFor
```

```
PaintBuffer
```

Questo programma visualizzerà tutti i 10 cerchi contemporaneamente.

Se il comando “UseBuffer” viene rimosso, ciascun cerchio sarà visualizzato come è disegnato.

Vedere anche: [PaintBuffer](#)

Elementi vuoti (nulli)

Quando si analizzano i dati del mondo reale, può accadere di non disporre sempre di una serie di dati completa. TI-Nspire™ consente l'uso di elementi vuoti o nulli. Ciò permette di proseguire con i dati a disposizione, anziché dover ricominciare da capo o scartare i casi incompleti.

Per un esempio di dati con elementi vuoti, vedere *"Rappresentazione grafica dei dati di un foglio di calcolo"* nel capitolo Foglio elettronico.

La funzione **delVoid()** consente di rimuovere elementi vuoti da una lista. Consente inoltre di testare un elemento vuoto. Per ulteriori informazioni, vedere **delVoid()**, pagina 42 e **isVoid()**, pagina 81.

Nota: per inserire manualmente un elemento vuoto in un'espressione matematica, digitare “_” o la parola chiave **void**. La parola chiave **void** viene convertita automaticamente in un carattere “_” quando l'espressione viene calcolata. Per digitare il carattere “_” sul palmare, premere **ctrl** **[]**.

Calcoli con elementi vuoti

La maggior parte dei calcoli con un inserimento vuoto producono un risultato vuoto. Vedere i casi speciali sotto.

$\lfloor \rfloor$	=
$\text{gcd}(100, \lfloor \rfloor)$	=
$3 + \lfloor \rfloor$	=
$\{5, \lfloor \rfloor, 10\} - \{3, 6, 9\}$	$\{2, \lfloor \rfloor, 1\}$

Argomenti di lista contenenti elementi vuoti

Le funzioni e i comandi seguenti ignorano (saltano) gli elementi vuoti che trovano negli argomenti di lista.

count, **countIf**, **cumulativeSum**, **freqTable**►list, **frequency**, **max**, **mean**, **median**, **product**, **stDevPop**, **stDevSamp**, **sum**, **sumIf**, **varPop** e **varSamp**, nonché i calcoli di regressione, le statistiche **OneVar**, **TwoVar** e **FiveNumSummary**, gli intervalli di confidenza e i test statistici.

$\text{sum}(\{2, \lfloor \rfloor, 3, 5, 6, 6\})$	16.6
$\text{median}(\{1, 2, \lfloor \rfloor, \lfloor \rfloor, 3\})$	2
$\text{cumulativeSum}(\{1, 2, \lfloor \rfloor, 4, 5\})$	$\{1, 3, \lfloor \rfloor, 7, 12\}$
$\text{cumulativeSum}\begin{bmatrix} 1 & 2 \\ 3 & \lfloor \rfloor \\ 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 \\ 4 & \lfloor \rfloor \\ 9 & 8 \end{bmatrix}$

SortA e **SortD** spostano in fondo tutti gli elementi vuoti del primo argomento.

$\{5, 4, 3, \lfloor \rfloor, 1\} \rightarrow \text{list1}$	$\{5, 4, 3, \lfloor \rfloor, 1\}$
$\{5, 4, 3, 2, 1\} \rightarrow \text{list2}$	$\{5, 4, 3, 2, 1\}$
SortA $\text{list1}, \text{list2}$	Done
list1	$\{1, 3, 4, 5, \lfloor \rfloor\}$
list2	$\{1, 3, 4, 5, 2\}$

Argomenti di lista contenenti elementi vuoti

Nelle regressioni, un elemento vuoto in una lista X o Y introduce un elemento vuoto corrispondente nel residuo.

$\{1,2,3,_,5\} \rightarrow list1$	$\{1,2,3,_,5\}$
$\{1,2,3,4,5\} \rightarrow list2$	$\{1,2,3,4,5\}$
SortD $list1, list2$	Done
$list1$	$\{5,3,2,1,_$
$list2$	$\{5,3,2,1,4\}$

$l1 := \{1,2,3,4,5\}; l2 := \{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
LinRegMx $l1, l2$	Done
$stat.Resid$	$\{0.434286,_, -0.862857, -0.011429, 0.44\}$
$stat.XReg$	$\{1,_, 3, 4, 5\}$
$stat.YReg$	$\{2,_, 3, 5, 6, 6\}$
$stat.FreqReg$	$\{1,_, 1, 1, 1, 1\}$

Una categoria omessa nelle regressioni introduce un elemento vuoto corrispondente nel residuo.

$l1 := \{1,3,4,5\}; l2 := \{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
$cat := \{"M", "M", "F", "F"\}; incl := \{"F"\}$	$\{"F"\}$
LinRegMx $l1, l2, cat, incl$	Done
$stat.Resid$	$\{_, _, 0, 0, _\}$
$stat.XReg$	$\{_, _, 4, 5, _\}$
$stat.YReg$	$\{_, _, 5, 6, 6\}$
$stat.FreqReg$	$\{_, _, 1, 1, _\}$

Una frequenza 0 nelle regressioni introduce un elemento vuoto corrispondente nel residuo.

$l1 := \{1,3,4,5\}; l2 := \{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx $l1, l2, \{1, 0, 1, 1\}$	Done
$stat.Resid$	$\{0.069231, _, -0.276923, 0.207692\}$
$stat.XReg$	$\{1, _, 4, 5, _\}$
$stat.YReg$	$\{2, _, 5, 6, 6\}$
$stat.FreqReg$	$\{1, _, 1, 1, _\}$

Scelte rapide per l'inserimento di espressioni matematiche

Le scelte rapide permettono di inserire elementi di espressioni matematiche per digitazione invece di usare il Catalogo o la tavolozza Simboli. Per esempio, per inserire l'espressione $\sqrt{6}$, è possibile digitare `sqrt(6)` nella riga di introduzione. Quando si preme **enter**, l'espressione `sqrt(6)` viene modificata in $\sqrt{6}$. Alcune scelte rapide sono utili sia per il palmare sia per la tastiera del computer. Altre sono utili principalmente dalla tastiera del computer.

Dalla tastiera del palmare o del computer

Per inserire questo:	digitare questa scelta rapida:
π	<code>pi</code>
θ	<code>theta</code>
∞	<code>infinity</code>
\leq	<code><=</code>
\geq	<code>>=</code>
\neq	<code>/=</code>
\Rightarrow (implicazione logica)	<code>=></code>
\Leftrightarrow (doppia implicazione logica, XNOR)	<code><=></code>
\rightarrow (memorizza operatore)	<code>=:</code>
$ $ (valore assoluto)	<code>abs(...)</code> (Valore assoluto)
$\sqrt()$	<code>sqrt(...)</code> (Radice quadrata)
$\Sigma()$ (Modello di somma)	<code>sumSeq(...)</code>
$\prod()$ (Modello di prodotto)	<code>prodSeq(...)</code>
$\sin^{-1}()$, $\cos^{-1}()$, ...	<code>arcsin(...), arccos(...), ...</code>
Δ lista()	<code>deltaList(...)</code>

Dalla tastiera del computer

Per inserire questo:	digitare questa scelta rapida:
i (unità immaginaria)	<code>@i</code>
e (logaritmo naturale in base e)	<code>@e</code>
E (notazione scientifica)	<code>@E</code>
T (trasposizione)	<code>@t</code>

Per inserire questo:	digitare questa scelta rapida:
r (radiani)	@r
$^\circ$ (Gradi)	@d
G gradianti	@g
\angle (angolo)	@<
► (conversione)	@>
►Decimal, ►approxFraction (), ecc.	@>Decimal, @>approxFraction(), ecc. (), ecc.

EOS™ (Equation Operating System) gerarchia

Questa sezione descrive il sistema EOS™ (Equation Operating System) utilizzato da TI-Nspire™ Sistema integrato per la matematica e le scienze sperimentaliste. Numeri, variabili e funzioni vengono introdotte in successione. Il software EOS™ calcola le espressioni e le equazioni utilizzando raggruppamenti racchiusi tra parentesi e in base alle priorità descritte sotto.

Ordine di Valutazione

Livello	Operatore
1	Parentesi tonde (), parentesi quadre [], parentesi graffe { }
2	Conversione indiretta (#)
3	Chiamate di funzione
4	Post operatori: gradi-primi-secondi($^{\circ}, ", !$), fattoriale (!), percentuale (%), radiante (Γ), pedice ([]), trasposizione (T)
5	Elevamento a potenza, operatore di potenza (^)
6	Segno negativo (-)
7	Concatenazione di stringhe (&)
8	Moltiplicazione (\cdot), divisione (/)
9	Addizione (+), sottrazione (-)
10	Rapporti di uguaglianza: uguale (=), non uguale (\neq o $/=$), minore di (<), minore di o uguale a (\leq o \leqslant), maggiore di (>), maggiore di o uguale a (\geq o \geqslant)
11	not logico
12	and logico
13	Logico or
14	xor, nor, nand
15	Implicazione logica (\Rightarrow)
16	Doppia implicazione logica, XNOR (\Leftrightarrow)
17	Operatore di vincolo (" ")
18	Memorizzazione (\rightarrow)

Parentesi tonde, quadre e graffe

Tutti i calcoli racchiusi tra parentesi tonde, quadre o graffe vengono eseguiti per primi. Ad esempio, nell'espressione $4(1+2)$, EOS™ calcola prima la parte di espressione racchiusa tra parentesi tonde, $1+2$, quindi moltiplica il risultato, 3 , per 4 .

In un'espressione o in un'equazione, tutte le parentesi tonde, quadre e graffe aperte devono essere chiuse. Diversamente, viene visualizzato un messaggio di errore a indicare l'elemento mancante. Ad esempio, $(1+2)/(3+4$ produrrà un messaggio di errore "Manca)."

Nota: poiché il software TI-Nspire™ consente di definire funzioni personalizzate, un nome di variabile seguito da un'espressione racchiusa tra parentesi viene considerato una "chiamata di funzione" e non una moltiplicazione implicita. Ad esempio $a(b+c)$ è la funzione a calcolata per $b+c$. Per moltiplicare l'espressione $b+c$ per la variabile a, utilizzare la moltiplicazione esplicita: $a*(b+c)$.

Conversione indiretta

L'operatore di conversione indiretta (#) converte una stringa in un nome di variabile o di funzione. Ad esempio, #("x"&"y"&"z") crea il nome di variabile xyz. La conversione indiretta consente inoltre di creare e modificare variabili dall'interno di un programma. Ad esempio, se $10\rightarrow r$ e $"r"\rightarrow s1$, allora $#s1=10$.

Post operatori

I post operatori sono operatori che vengono inseriti immediatamente dopo un argomento, come ad esempio $5!$, 25% o $60^{\circ}15'45''$. Gli argomenti seguiti da un post operatore vengono calcolati al quarto livello di priorità. Ad esempio, nell'espressione $4^3!$, $3!$ viene calcolato per primo. Il risultato, 6, diventa quindi l'esponente di 4 che darà come risultato 4096.

Elevazione a potenza

L'elevamento a potenza (^) e l'elevazione a potenza elemento per elemento (.^) vengono calcolati da destra a sinistra. Ad esempio, l'espressione 2^3^2 viene calcolata allo stesso modo di $2^(3^2)$ dando come risultato 512. Ciò è diverso da $(2^3)^2$, che dà come risultato 64.

Segno negativo

Per introdurre un numero negativo, premere [(-)] quindi il numero. Le post operazioni e l'elevamento a potenza vengono eseguiti prima dell'operazione di cambiamento di segno. Ad esempio, il risultato di $-x^2$ è un numero negativo e $-9^2 = -81$. Utilizzare le parentesi per elevare al quadrato un numero negativo, come ad esempio $(-9)^2$ che dà come risultato 81.

Vincolo ("|")

L'argomento che segue l'operatore di vincolo ("|") fornisce una serie di restrizioni che influiscono sul calcolo dell'argomento che precede l'operatore.

TI-Nspire CX II - Funzioni di programmazione di TI-Basic

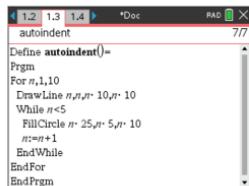
Rientro automatico in Program Editor

Program Editor di TI-Nspire™ esegue ora il rientro automatico delle istruzioni all'interno di un comando di blocco.

I comandi di blocco sono If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry.

L'editor anteporrà automaticamente gli spazi per programmare i comandi all'interno di un comando di blocco. Il comando di chiusura del blocco verrà allineato con il comando di apertura.

Di seguito viene mostrato un esempio di rientro automatico in comandi di blocco nidificati.



```
1.2 | 1.3 | 1.4 *Dot PAD X
autoident
Define autoident()=
Prgm
For n,1,10
DrawLine n,n,n- 10,n- 10
While n<5
FillCircle n- 25,n- 5,n- 10
n:=n+1
EndWhile
EndFor
EndPrgm
```

I frammenti di codice copiati e incollati manterranno il rientro originale.

Se si apre un programma creato in una versione precedente del software, verrà mantenuto il rientro originale.

Miglioramento dei messaggi di errore per TI-Basic

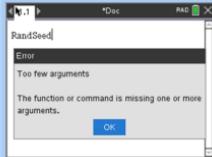
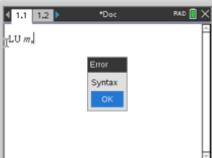
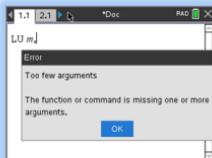
Errori

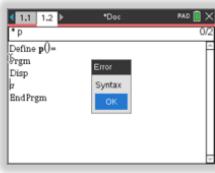
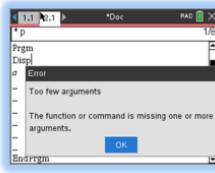
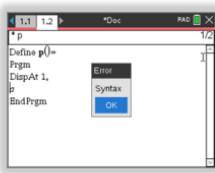
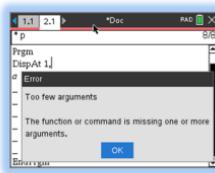
Condizione di errore	Nuovo messaggio
Errore nell'istruzione condizionale (If/While)	Un'istruzione condizionale non ha restituito VERO o FALSO NOTA: con la modifica che stabilisce di posizionare il cursore sulla linea con l'errore, non è più necessario specificare se l'errore si trova in un'istruzione "If" o in un'istruzione "While".
EndIf mancante	Trovata istruzione end diversa dall'istruzione EndIf attesa
EndFor mancante	Trovata istruzione end diversa dall'istruzione EndFor attesa
EndWhile mancante	Trovata istruzione end diversa dall'istruzione EndWhile attesa

Condizione di errore	Nuovo messaggio
EndLoop mancante	Trovata istruzione end diversa dall'istruzione EndLoop attesa
EndTry mancante	Trovata istruzione end diversa dall'istruzione EndTry attesa
"Then" omessa dopo If <condition>	If..Then mancante
"Then" omessa dopo ElseIf <condition>	Manc Then nel blocco: ElseIf .
Quando sono stati incontrate "Then", "Else" ed "ElseIf" al di fuori dei blocchi di controllo	Else non valida fuori dai blocchi: If..Then..Endif o Try..EndTry
"ElseIf" appare al di fuori del blocco "If..Then..Endif"	ElseIf non valida fuori dal blocco: If..Then..Endif
"Then" appare al di fuori del blocco "If....Endif"	Then non valida fuori dal blocco: If..Endif

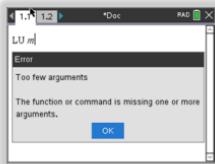
Errori di sintassi

Nel caso in cui dei comandi che devono contenere uno o più argomenti vengono chiamati con un elenco di argomenti incompleto, verrà generato il messaggio di errore **"Argomenti mancanti"** invece di un errore **"Sintassi"**.

Comportamento corrente	Nuovo comportamento CX II
	
	

Comportamento corrente	Nuovo comportamento CX II
	
	

Nota: quando un elenco di argomenti incompleto non viene seguita da una virgola, viene visualizzato il messaggio di errore: "Argomenti mancanti". Ciò si verifica anche nelle versioni precedenti.



Costanti e valori

Nella seguente tabella sono elencate le costanti con i rispettivi valori disponibili durante l'esecuzione delle conversioni di unità. Possono essere digitate manualmente o selezionate dall'elenco **Costanti** in **Utilità > Conversioni di unità** (palmare: premere 3).

Costante	Nome	Valore
_c	Velocità della luce	299792458 _m/_s
_Cc	Costante di coulomb	8987551787.3682 _m/_F
_Fc	Costante di Faraday	96485.33289 _coul/_mol
_g	Accelerazione di gravità	9.80665 _m/_s ²
_Gc	Costante gravitazionale	6.67408E-11 _m ³ /kg/_s ²
_h	Costante di Planck	6.626070040E-34 _J_s
k	Costante di Boltzmann	1.38064852E-23 J/°K
_μ0	Permeabilità del vuoto	1.2566370614359E-6 _N/_A ²
_μb	Magnetone di Bohr	9.274009994E-24 _J_m ² /_Wb
_Me	Massa a riposo dell'elettrone	9.10938356E-31_kg
_Mμ	Massa del muone	1.883531594E-28_kg
_Mn	Massa a riposo del neutrone	1.674927471E-27_kg
_Mp	Massa a riposo del protone	1.672621898E-27_kg
_Na	Numero di Avogadro	6.022140857E23 /_mol
_q	Carica dell'elettrone	1.6021766208E-19 _coul
_Rb	Raggio di bohr	5.2917721067E-11 _m
_Rc	Costante molare del gas	8.3144598 _J/_mol/_°K
_Rdb	costante di Rydberg	10973731.568508/_m
_Re	Raggio dell'elettrone	2.8179403227E-15 _m
_u	Massa atomica	1.660539040E-27_kg
_Vm	Volume molare	2.2413962E-2 _m ³ /_mol
_ε0	Permittività del vuoto	8.8541878176204E-12 _F/_m
_σ	Costante di Stefan-Boltzmann	5.670367E-8 _W/_m ² /_°K ⁴
_Φ0	Quanto di flusso magnetico	2.067833831E-15 _Wb

Codici di errore e messaggi

Quando si produce un errore, il relativo codice viene assegnato alla variabile *errCode*. Programmi e funzioni definite dall'utente possono esaminare *errCode* per determinare la causa dell'errore. Per un esempio dell'uso di *errCode*, vedere l'Esempio 2 del comando Try (pagina 172).

Nota: alcune condizioni di errore si riferiscono solo ai prodotti TI-Nspire™ CAS, mentre altre solo ai prodotti TI-Nspire™.

Codice errore	Descrizione
10	Una funzione non ha restituito un valore
20	Una prova non ha saputo stabilire se VERO o FALSO. In generale, non è possibile confrontare variabili non definite. Ad esempio, la prova If $a < b$ causerà questo errore se a o b sono indefiniti al momento dell'esecuzione dell'istruzione If.
30	Argomento non può essere un nome di cartella.
40	Argomento errato
50	Argomento di tipo errato Due o più argomenti devono essere dello stesso tipo.
60	Argomento deve essere un'espressione booleana o un numero intero
70	Argomento deve essere un numero decimale
90	Argomento deve essere una lista
100	Argomento deve essere una matrice
130	Argomento deve essere una stringa
140	Argomento deve essere un nome di variabile. Accertarsi che il nome: <ul style="list-style-type: none">• non inizi con un numero• non contenga spazi o caratteri speciali• non utilizzi trattini bassi o virgolette in modo non valido• non superi i limiti di lunghezza Per ulteriori informazioni, vedere il capitolo Calcolatrice nella documentazione.
160	Argomento deve essere un'espressione
165	Carica delle batterie insufficiente per inviare/ricevere Inserire nuove batteria prima di inviare o ricevere.
170	Estremo

Codice errore	Descrizione
	L'estremo inferiore deve essere minore dell'estremo superiore per definire l'intervallo di ricerca.
180	Interruzione È stato premuto il tasto [esc] o [fn on] durante un calcolo lungo o durante l'esecuzione del programma.
190	Definizione circolare Questo messaggio viene visualizzato per evitare l'esaurimento della memoria durante la sostituzione infinita di valori di variabile nel corso di una semplificazione. Ad esempio, $a+1>a$, dove a è una variabile indefinita, causerà questo errore.
200	Condizione non valida Ad esempio, $\text{solve}(3x^2-4=0,x) \mid x<0 \text{ or } x>5$ produrrebbero questo messaggio di errore perché il vincolo è separato da "or" invece che da "and."
210	Tipo di dati non valido Un argomento è un tipo di dati sbagliato.
220	Limite dipendente
230	Dimensione Un indice di lista o di matrice non è valido. Ad esempio, se la lista {1,2,3,4} è memorizzata in L1, allora L1[5] è un errore di dimensione perché L1 contiene solo quattro elementi.
235	Errore di dimensione. Elementi insufficienti nelle liste.
240	Dimensioni non corrispondenti Due o più argomenti devono essere della stessa dimensione. Ad esempio, [1,2]+[1,2,3] è un errore di dimensioni non corrispondenti perché le matrici contengono un numero diverso di elementi.
250	Divisione per zero
260	Argomento errato Deve essere presente un argomento in un dominio specificato. Ad esempio rand(0) non è valido.
270	Nome di variabile duplicato
280	Else ed Elself non validi fuori dal blocco If..Endif
290	EndTry non trova la corrispondente istruzione Else
295	Numero eccessivo di iterazioni

Codice errore	Descrizione
300	Attesa lista o matrice a 2 o 3 elementi
310	Il primo argomento di nSolvedeve essere un'equazione a una sola variabile. Non può contenere una variabile non calcolata che non sia la variabile cercata.
320	Il primo argomento di solve o cSolve deve essere un'equazione o una disequazione. Ad esempio, solve($3x^2-4,x$) non è valido perché il primo argomento non è un'equazione.
345	Unità di misura non coerenti
350	Indice non valido
360	La stringa da convertire non è un nome di variabile valido
380	Risultato indefinito Il precedente calcolo non ha prodotto un risultato oppure non è stato inserito alcun calcolo precedente.
390	Assegnazione non valida
400	Valore di assegnazione non valido
410	Comando non valido
430	Non valido per le impostazioni di modo correnti
435	Ipotesi non valida
440	Moltiplicazione sottintesa non valida Ad esempio, $x(x+1)$ non è valido, mentre $x*(x+1)$ è la sintassi corretta. Ciò per evitare confusione tra chiamate di funzione e moltiplicazioni implicite.
450	Non valido in una funzione o nella attuale espressione In una funzione definita dall'utente sono validi solo certi comandi.
490	Non valido nel blocco Try..EndTry
510	Lista o matrice non valida
550	Non valido fuori da una funzione o un programma Alcuni comandi non sono validi fuori da una funzione o un programma. Ad esempio, Local non può essere utilizzato in una funzione o in un programma.
560	Non valido fuori dai blocchi Loop..EndLoop, For..EndFor o While..EndWhile Ad esempio, il comando Exit è valido solo in questi blocchi loop.
565	Non valido fuori da un programma
570	Nome di percorso non valido

Codice errore	Descrizione
	Ad esempio, \var non è valido.
575	Numero complesso in forma polare non valido
580	Chiamata di programma non valida Non è possibile chiamare un programma all'interno di funzioni o espressioni quali $1+p(x)$, dove p è un programma.
600	Tabella non valida
605	Uso di unità non valido
610	Nome di variabile non valido in una istruzione Local
620	Nomi di variabile o di funzione non valido
630	Chiamata di variabile non valida
640	Sintassi del vettore non valida
650	Errore di collegamento Trasmissione non completata tra due unità. Verificare che il cavo di collegamento sia inserito correttamente ad entrambe le estremità.
665	Matrice non diagonalizzabile
670	Memoria quasi esaurita 1. Eliminare alcuni dati in questo documento 2. Salvare e chiudere questo documento Se le istruzioni 1 e 2 non producono l'esito sperato, togliere e rimettere le batterie
672	Risorsa esaurita
673	Risorsa esaurita
680	Manca (
690	Manca)
700	Manca "
710	Manca]
720	Manca }
730	Manca inizio o fine del blocco
740	Manca Then nel blocco If..EndIf
750	Il nome non è una funzione o un programma

Codice errore	Descrizione
765	Nessuna funzione selezionata
780	Nessuna soluzione trovata
800	Risultato non reale Ad esempio, se il software è impostato su Real, $\sqrt{(-1)}$ non è valido. Per consentire risultati complessi, modificare l'impostazione di modalità "Real or Complex" in RETTANGOLARE (RECTANGULAR) o POLARE (POLAR).
830	Superamento della memoria
850	Programma non trovato Impossibile trovare una chiamata di programma all'interno di un altro programma nel percorso specificato durante l'esecuzione.
855	Funzione di tipo Rand non consentita nel grafico
860	Ricorsione troppo profonda
870	Nome riservato o variabile di sistema
900	Argomento errato Impossibile applicare al set di dati il modello mediana-mediana.
910	Errore di sintassi
920	Testo non trovato
930	Argomenti mancanti Nella funzione o nel comando mancano uno o più argomenti.
940	Troppi argomenti L'espressione o equazione contiene un numero eccessivo di argomenti e non può essere calcolata.
950	Troppi indici
955	Troppe variabili non definite
960	La variabile non è definita Nessun valore assegnato alla variabile. Utilizzare uno dei seguenti comandi: <ul style="list-style-type: none">• sto \rightarrow• :=• Define per assegnare valori alle variabili.

Codice errore	Descrizione
965	SO privo di licenza
970	Variabile in uso, di conseguenza non sono consentite chiamate e modifiche
980	La variabile è protetta
990	Nome variabile non valido Accertarsi che il nome non superi i limiti di lunghezza
1000	Valori dei parametri di Window
1010	Zoom
1020	Errore interno
1030	Violazione della memoria protetta
1040	Funzione non supportata. Questa funzione richiede il sistema CAS (Computer Algebra System). Provare TI-Nspire™ CAS.
1045	Operatore non supportato. Questo operatore richiede il sistema CAS (Computer Algebra System). Provare TI-Nspire™ CAS.
1050	Funzione non supportata. Questo operatore richiede il sistema CAS (Computer Algebra System). Provare TI-Nspire™ CAS.
1060	L'argomento dell'inserimento deve essere numerico. Sono consentiti solo inserimenti contenenti valori numerici.
1070	L'argomento della funzione trigonometrica è troppo grande per una riduzione precisa.
1080	Uso non supportato di Ans. Questa applicazione non supporta Ans.
1090	La funzione non è definita. Utilizzare uno dei seguenti comandi: <ul style="list-style-type: none">• Define• :=• sto \rightarrow per definire una funzione.
1100	Calcolo non reale Ad esempio, se il software è impostato su "Real" $\sqrt{(-1)}$ non è valido. Per consentire risultati complessi, modificare l'impostazione di modalità "Real or Complex" in RETTANGOLARE (RECTANGULAR) o POLARE (POLAR).
1110	Estremi non validi
1120	Nessun cambio di segno

Codice errore	Descrizione
1130	Argomento non può essere una lista o una matrice
1140	<p>Argomento errato</p> <p>Il primo argomento deve essere un'espressione polinomiale nel secondo argomento. Se il secondo argomento viene omesso, il software tenta di selezionare un'impostazione predefinita.</p>
1150	<p>Argomento errato</p> <p>I primi due argomenti devono essere espressioni polinomiali nel terzo argomento. Se il terzo argomento viene omesso, il software tenta di selezionare un'impostazione predefinita.</p>
1160	<p>Nome di percorso libreria non valido</p> <p>Un nome di percorso deve avere la forma <code>xxx\yyy</code>, dove:</p> <ul style="list-style-type: none"> • La parte <code>xxx</code> può avere da 1 a 16 caratteri. • La parte <code>yyy</code> può avere da 1 a 15 caratteri. <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1170	<p>Uso di nome percorso libreria non valido</p> <ul style="list-style-type: none"> • Impossibile assegnare un valore a un nome percorso utilizzando Define, <code>:=</code> o sto \rightarrow. • Impossibile dichiarare un nome percorso come variabile Local oppure utilizzarlo come parametro in una definizione di funzione o programma.
1180	<p>Nome variabile libreria non valido.</p> <p>Accertarsi che il nome:</p> <ul style="list-style-type: none"> • non contenga un punto • non inizi con un trattino basso • non contenga più di 15 caratteri <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1190	<p>Documento Libreria non trovato:</p> <ul style="list-style-type: none"> • Verificare che la libreria sia nella cartella MieLibrerie. • Aggiornare le librerie. <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1200	<p>Variabile libreria non trovata:</p> <ul style="list-style-type: none"> • Verificare che la variabile libreria esista nella prima attività nella libreria. • Accertarsi che la variabile libreria sia stata definita come LibPub o LibPriv.

Codice errore	Descrizione
	<ul style="list-style-type: none"> • Aggiornare le librerie. <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1210	<p>Nome collegamento libreria non valido.</p> <p>Accertarsi che il nome:</p> <ul style="list-style-type: none"> • non contenga un punto • non inizi con un trattino basso • non contenga più di 16 caratteri • non sia un nome riservato <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1220	<p>Errore di dominio.</p> <p>Le funzioni Retta tangente e Retta normale supportano solo funzioni con valori reali.</p>
1230	<p>Errore di dominio.</p> <p>Gli operatori di conversione trigonometrica non sono supportati nelle modalità in gradi o gradianti.</p>
1250	<p>Argomento errato</p> <p>Utilizza un sistema di equazioni lineari.</p> <p>Esempio di sistema di due equazioni lineari con variabili x e y:</p> $3x+7y=5$ $2y-5x=-1$
1260	<p>Argomento errato:</p> <p>Il primo argomento di nfMin o nfMax deve essere un'espressione a una sola variabile. Non può contenere una variabile non calcolata che non sia la variabile cercata.</p>
1270	<p>Argomento errato</p> <p>L'ordine della derivata deve essere uguale a 1 o 2.</p>
1280	<p>Argomento errato</p> <p>Utilizzare un polinomio in forma normale in una sola variabile.</p>
1290	<p>Argomento errato</p> <p>Utilizzare un polinomio in una sola variabile.</p>
1300	<p>Argomento errato</p> <p>I coefficienti del polinomio devono dare come risultato valori numerici.</p>

Codice errore	Descrizione
1310	Argomento errato: Non è stato possibile calcolare una funzione per uno o più dei suoi argomenti.
1380	Argomento errato: Non sono autorizzate le chiamate nidificate alla funzione domain().

Codici di avvertenza e messaggi

È possibile utilizzare la `warnCodes()` funzione per memorizzare i codici delle avvertenze generate dal calcolo di un'espressione. Questa tabella riporta ciascun codice numerico e il relativo messaggio associato. Per un esempio di memorizzazione dei codici di avvertenza, vedere `warnCodes()`, pagina 181.

Codice di avvertenza	Messaggio
10000	L'operazione può introdurre soluzioni false Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10001	Derivare un'equazione può produrre un'equazione falsa
10002	Soluzione dubbia Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10003	Accuratezza dubbia Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10004	L'operazione può far perdere soluzioni Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10005	cSolve può specificare più zeri
10006	Solve può specificare più zeri Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10007	È possibile che ci siano altre soluzioni. Provare a definire i valori inferiori e superiori adeguati e/o una ipotesi. Esempi che usano solve(): <ul style="list-style-type: none">• <code>solve(Equation, Var=Approssimativo) lowBound<Var<upBound</code>• <code>solve(Equation, Var) lowBound<Var<upBound</code>• <code>solve(Equation,Var=Approssimativo)</code> Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10008	Il dominio del risultato può essere minore del dominio dell'introduzione.
10009	Il dominio del risultato può essere maggiore del dominio dell'introduzione.
10012	Calcolo non reale
10013	∞^0 o indef^0 sostituito da 1
10014	indef^0 sostituito da 1
10015	1^∞ o 1^{indef} sostituito da 1

Codice di avvertenza	Messaggio
10016	1 [^] indef sostituito da 1
10017	Overflow sostituito da ∞ o $-\infty$
10018	L'operazione richiede e restituisce un valore a 64 bit
10019	Risorsa esaurita, semplificazione forse incompleta
10020	L'argomento della funzione trigonometrica è troppo grande per una riduzione precisa
10021	L'introduzione contiene un parametro non definito. Il risultato potrebbe non essere valido per tutti i valori di parametro possibili.
10022	La specifica di estremi inferiore e superiore appropriati può produrre una soluzione.
10023	Lo scalare è stato moltiplicato per la matrice identità.
10024	Risultato ottenuto utilizzando aritmetica approssimata.
10025	L'equivalenza non può essere verificata in modalità EXACT (Esatta).
10026	È possibile ignorare il vincolo. Specificare un vincolo nella forma "\[Costante variabile MathTestSymbol]" oppure un insieme di queste forme, ad esempio 'x<3 and x>-12'

Informazioni Generali

Guida online

education.ti.com/eguide

Selezionare il proprio Paese per maggiori informazioni sul prodotto.

Contattare l'assistenza TI

education.ti.com/ti-cares

Selezionare il proprio Paese per assistenza tecnica e altre risorse.

Informazioni su servizi e garanzia

education.ti.com/warranty

Selezionare il proprio Paese per informazioni sulla durata e i termini della garanzia oppure sui servizi per i prodotti.

Garanzia limitata. La presente garanzia non pregiudica i diritti spettanti per legge.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Indice

-		^, elevamento a potenza	194
-		-, sottrazione[*]	191
!		!, operatore di vincolo	212
!, fattoriale	202		
"		+ , addizione	191
", notazione in secondi	210	/ , divisione[*]	193
#		=	
#, conversione indiretta	207	#, diverso[*]	198
#, operatore conversione indiretta	238	=, uguale	198
%		>	
%, percentuale	197	>, maggiore di	200
&		\prod	
&, aggiunge	202	\prod , prodotto[*]	204
*		Σ	
*		$\Sigma()$, somma[*]	205
,		$\Sigma\text{Int}()$	205
, notazione in primi	210	$\Sigma\text{Prn}()$	206
,		\sqrt	
.-, punto sottrazione	196	\sqrt , radice quadrata[*]	203
.*, punto moltiplicazione	196	\int	
./, punto divisione	196	\int , integrale[*]	203
.^, punto elevato a potenza	197		
.+, punto addizione	195	\leq	
:		\leq , minore di o uguale a	199
:=, assegna	214		
\wedge			
\wedge^{-1} , reciproco	212		

\geq		$^{\circ}$, notazione in gradi[*]	209
\geq , maggiore di o uguale a	200	0	
►		0b, indicatore binario	214
►, converti in angolo in gradianti [Grad]	73	0h, indicatore esadecimale	214
►approxFraction()	13	1	
►Base10, visualizza come numero decimale[Base10]	18	10^(), potenza di dieci	211
►Base16, visualizza come esadecimale [Base16]	19	A	
►Base2, visualizza come binario [Base2]	17	abs(), valore assoluto	7
►Cylind, visualizza come vettore in forma cilindrica[Cylind] ...	37	addizione, +	191
►DD, visualizza angolo decimale[DD]	38	affianca(concatena, augment())	15
►Decimal, visualizza il risultato nella forma decimale[Decimal] ..	38	aggiunge, &	202
►DMS, visualizza come gradi/primi/secondi[DMS] ..	46	all'interno della stringa, inString() ..	77
►Polar, visualizza come vettore polare[Polar]	121	altrimenti, Else	74
►Rad, converti in angolo in radianti ..	129	amortTbl(), tabella di ammortamento	7, 16
►Rect, visualizza come vettore rettangolare	133	and, operatore boolean	8
►Sphere, visualizza come vettore sferico[Sphere]	158	angle(), angolo	9
⇒		angolo, angle()	9
⇒, implicazione logica[*]	201, 235	ANOVA, analisi della varianza a una variabile	9
→		ANOVA2way, analisi della varianza a due dimensioni	10
→, memorizza	213	Ans, ultimo risultato	12
↔		approssima, approx()	13
↔, doppia implicazione logica[*] ...	201	approx(), approssima	13
©		approxRational()	13
©, commento	214	arccos()	14
°		arccosh()	14
°, gradi/primi/secondi[*]	210	arccot()	14
		arccoth()	14
		arccsc()	14
		arccsch()	14
		arcocoseno, $\cos^{-1}()$	28
		arcoseno, $\sin^{-1}()$	154
		arcotangente, $\tan^{-1}()$	166
		arcsec()	14
		arcsech()	14
		arcsin()	14
		arcsinh()	14

arctan()	15	matrice	
arctanh()	15	colNorm(), norma colonna di matrice	25
argomenti nelle funzioni TVM	176	comando Stop	163
argomenti TVM	176	Comando Wait	180
arrotondamento, round()	142	combinazioni, nCr()	107
arrotondato per difetto, floor()	57	commento, ©	214
arrotondato per eccesso, ceiling()	20-21, 33	complesso coniugato, conj()	25
augment(), affianca/concatena	15	con, 	212
autovalore, eigVl()	48	conj(), complesso coniugato	25
autovettore, eigVc()	48	constructMat(), Costruisci matrice ..	26
avgRC(), tasso di variazione media ..	15	conteggio condizionato elementi in una lista, countif()	32
B			
binario		conteggio elementi in una lista, count()	32
indicatore, 0b	214	conteggio giorni tra le date, dbd()	37
visualizza, ►Base2	17	conversione indiretta, #	207
binomCdf() (Funzione della probabilità cumulativa per la distribuzione binomiale)	20	converti ►Rad	129
binomPdf()	20	4Grad	73
blocco di variabile e gruppi di variabili	93	coordinata polare, R►Pr()	129
C			
calcola polinomio, polyEval()	122	coordinata polare, R►Pθ()	129
calcolo, ordine di	237	coordinata x rettangolare, P►Rx()	118
campione casuale	132	coordinata y rettangolare, P►Ry()	119
cancella		copia variabile o funzione, CopyVar ..	26
errore, ClrErr	24	corrMat(), matrice di correlazione ..	27
Cancellare	220	cos ⁻¹ , arcocoseno	28
caratteri		cos(), coseno	27
codice di carattere, ord()	118	coseno, cos()	27
stringa, char()	21	cosh ⁻¹ (), arcocoseno iperbolico ..	30
Cdf()	55	cosh(), coseno iperbolico	29
ceiling(), arrotondato per eccesso ..	20	Costruisci matrice, constructMat()	26
centralDiff()	21	cot ⁻¹ (), arcocotangente	31
char(), stringa di caratteri	21	cot(), cotangente	30
χ ² way	22	cotangente, cot()	30
ClearAZ	24	coth ⁻¹ (), arcocotangente iperbolica ..	31
ClrErr, cancella errore	24	coth(), cotangente iperbolica	31
codici di avvertenza e messaggi	252	count(), conteggio elementi in una lista	32
Codici di errore e messaggi	252	countif(), conteggio condizionato elementi in una lista	32
colAugment	25	cPolyRoots()	33
colDim(), dimensione colonna di	25	crossP(), prodotto vettoriale	33

csc ⁻¹ (), cosecante inversa	34	dim(), dimensione	43
csc(), cosecante	34	dimensione, dim()	43
csch ⁻¹ (), cosecante iperbolica inversa	35	dimensioni di righe di matrice, rowDim()	143
csch(), cosecante iperbolica	34	disegna	221-223
CubicReg, regressione cubica	35	Disp, visualizza dati	44
cumulativeSum(), somma cumulativa	36	DispAt	44
Cycle, ripeti	36	disposizioni semplici, nPr()	114
		distribuzione normale cumulativa inversa (invNorm())	80
		diverso, ≠	198
D		divisione intera, intDiv()	77
d(), prima derivata	202	divisione, P	193
da lista a matrice, list►mat()	90	doppia implicazione logica, ⇔	201
da matrice a lista, mat►list()	97	dot	
dbd(), giorni tra le date	37	prodotto, dotP()	46
decimale visualizza angolo, ►DD	38	dotP(), prodotto scalare	46
visualizza intero, ►Base10	18		
Define	39	E	
Define LibPriv	40	e alla potenza, e^()	52
Define LibPub	40	E, esponente	207
Define, definisci	39	e^(), funzione esponenziale	47
definisci, Define	39	eff), converti da tasso nominale a effettivo	47
definizione funzione o programma privato ..	40	eigVc(), autovettore	48
programma o funzione pubblica ..	40	eigVl(), autovalore	48
deltaList()	41	elementi vuoti	233
DelVar, elimina variabile	41	elementi vuoti (null)	233
delVoid(), rimuovi elementi vuoti ..	42	elementi vuoti, rimuovi	42
densità di probabilità t di Student, tPdf()	171	elevamento a potenza, ^	194
densità di probabilità, normPdf() ..	113	elimina elementi vuoti dalla lista	42
derivata prima modello di	5	eliminazione variabile, DelVar	41
derivata seconda modello di	6	else if, ElseIf	49
derivate derivata numerica, nDeriv() ...	109-110	ElseIf, else if	49
derivata numerica, nDerivative()	108	end	
prima derivata, d()	202	loop, EndLoop	96
destra, right()	78, 139	per, EndFor	58
det(), determinante matrice	42	EndTry, fine tentativo	172
deviazione standard, stdDev() ..	161-162, 179	EndWhile, fine mentre	182
diag(), diagonale matrice	43	EOS (Equation Operating System) ..	237
		Equation Operating System (EOS) ..	237

equazioni simultanee, simult()	153	forma a scalini ridotta per righe, rref()	144
errori e soluzione dei problemi		format(), stringa formato	58
cancella errore, ClrErr	24	fpart(), funzione parte frazionaria	59
passa errore, PassErr	119	frazione propria, propFrac	125
esadecimale		frazioni	
indicatore, Oh	214	frzProp	125
visualizza, ►Base16	19	modello di	1
esci, Exit	52	freqTable()	60
esclusione con l'operatore "!"	212	frequency()	60
esponente, E	207	Func, funzione	62
esponenti		Func, funzione programma	62
modello di	1	funzione esponenziale e	
espressioni		modello di	2
da stringa a espressione, expr()	53	funzione esponenziale, e^()	47
etichetta, Lbl	82	funzione piecewise a 2 tratti	
euler(), Euler function	50	modello di	2
Exit, esci	52	funzione piecewise a N tratti	
exp(), e alla potenza	52	modello di	3
expr(), da stringa a espressione	53	funzioni	
ExpReg, regressione esponenziale	53	definite dall'utente	39
		funzione programma, Func	62
F		parte frazionaria, fpart()	59
F Test su due campioni	61	funzioni definite dall'utente	39
factor(), fattorizza	55	funzioni di distribuzione	
fattoriale, !	202	binomCdf()	79
fattorizza, factor()	55	binomCdf()(Funzione della	
Fdrbinom()	79	probabilità cumulativa	
Fill, riempì matrice	56	per la distribuzione	
fine		binomiale)	20
funzione, EndFunc	62	binomPdf()	20
mentre, EndWhile	182	invNorm()	80
tentativo, EndTry	172	invt()	80
fine funzione, EndFunc	62	InvX ² ()	78
fine loop, EndLoop	96	normCdf()	112
fine mentre, EndWhile	182	normPdf()	113
fine se, EndIf	74	poissCdf()	120
FiveNumSummary	56	poissPdf()	121
floor(), arrotondato per difetto	57	tCdf()	168
For	58	tPdf()	171
For, per	58	χ ² 2way()	22
forma a scalini per righe, ref()	134	χ ² Cdf()	22
forma a scalini ridotta per righe		χ ² GOF()	23
matrici, rref()	144	χ ² Pdf()	24

funzioni e programmi definiti			
dall'utente	40	imposta	
funzioni e variabili		modo, setMode()	149
copia	26	impostazioni di modo, getMode() ..	70
funzioni finanziarie, tvmFV()	175	impostazioni, visualizza correnti ..	70
funzioni finanziarie, tvmI()	175	in mezzo alla stringa, mid()	101
funzioni finanziarie, tvmN()	175	indice interno di rendimento	
funzioni finanziarie, tvmPmt()	176	modificato, mirr()	102
funzioni finanziarie, tvmPV()	176	inString(), all'interno della stringa ..	77
		int(), intero	77
		intDiv(), divisione intera	77
		integrale definito	
		modello di	6
G		integrale, \int	203
G, gradianti	208	intero, int()	77
gcd(), massimo comun divisore	63	interpolate(), interpolare	78
geomCdf()	63	inversa, ${}^{-1}$	212
geomPdf()	64	inverti righe matrice, rowSwap()	144
Get	64, 227	invF()	79
getDenom(), ottieni/restituisci		invNorm(), distribuzione normale	
denominatore	65	cumulativa inversa	80
getKey()	65	invvt()	80
getLangInfo(), ottieni/restituisci		Invx 2 ()	78
informazioni sulla lingua	69	iPart(), parte intera	80
getLockInfo(), testa lo stato		iperbolico	
bloccato di una variabile o di		arcocoseno, cosh $^{-1}$ ()	30
un gruppo di variabili	69	arcoseno, sinh $^{-1}$ ()	156
getMode(), impostazioni di modo	70	arcotangente, tanh $^{-1}$ ()	168
getNum(), ottieni/restituisci		coseno, cosh()	29
numeratore	71	seno, sinh()	155
GetStr	71	tangente, tanh()	167
getType(), get type of variable	72	irr(), indice di rendimento interno	
getVarInfo(), ottieni/restituisci		indice di rendimento interno, irr	
informazioni variabile	72	()	80
giorni tra le date, dbd()	37	isPrime(), verifica numero primo	81
Goto, vai a	73	isVoid(), test per nullità	81
gruppi, blocco e sblocco	93, 179		
gruppi, test stato di blocco	69		
I		L	
identity(), matrice identità	74	lbl, etichetta	82
If, se	74	lcm, minimo comune multiplo	82
ifFn()	75	left(), sinistra	82
imag(), parte non reale	76	LibPriv	40
implicazione logica, \Rightarrow	201, 235	LibPub	40
		libreria	
		crea collegamenti a oggetti	83

libShortcut(), crea collegamenti a oggetti librerie	83	logaritmo naturale, ln()	90
linea di regressione mediana- mediana, MedMed	99	Logistic, regressione logistica	94
lingua		LogisticD, regressione logistica	95
ottiene informazioni sulla lingua	69	loop, Loop	96
LinRegBx, regressione lineare	83	Loop, loop	96
LinRegMx, regressione lineare	84	LU, scomposizione matrice inferiore- superiore	97
LinRegtIntervals, regressione lineare	86	lunghezza della stringa	43
LinRegtTest (t Test regressione lineare)	87	M	
linSolve()	89	maggiore di o uguale a, 	200
list►mat(), da lista a matrice	90	maggiore di, >	200
lista, conteggio condizionato	32	massimo comune divisore, gcd() ...	63
lista, conteggio elementi	32	massimo, max()	98
liste		mat►list(), da matrice a lista	97
affianca/concatena, augment()	15	matrice (1×2)	
da lista a matrice, list►mat() ...	90	modello di	4
da matrice a lista, mat►list() ...	97	matrice (2×1)	
differenza, @list()	89	modello di	4
differenze in una lista, @list()	89	matrice (2×2)	
elementi vuoti in	233	modello di	4
in mezzo alla stringa, mid()	101	matrice (m × n)	
massimo, max()	98	modello di	4
minimo, min()	101	matrice casuale, randMat()	131
nuova, newList()	109	matrice di correlazione, corrMat()	27
ordinamento ascendente, SortA		matrice identità, identity()	74
ordinamento discendente,		matrici	
SortD	158	affianca/concatena, augment()	15
prodotto scalare, dotP()	46	autovalore, eigVl()	48
prodotto vettoriale, crossP()	33	autovettore, eigVc()	48
prodotto, product()	124	da lista a matrice, list►mat()	90
somma cumulativa,		da matrice a lista, mat►list()	97
cumulativeSum()	36	determinante, det()	42
somma, sum()	164	diagonale, diag()	43
In(), logaritmo naturale	90	dimensione colonna, colDim()	25
LnReg, regressione logaritmica	91	dimensione, dim()	43
Local, variabile locale	92	forma a scalini per righe matrici,	
locale, Local	92	ref()	134
Lock, blocca variabile o gruppo di variabili	93	massimo, max()	98
Log		minimo, min()	101
modello di	2	moltiplicazione e somma di righe, mRowAdd()	103
logaritmi	90	norma colonna, colNorm()	25
		nuova, newMat()	109

operazione con righe, mRow() .	103	Log	2
prodotto, product()	124	matrice (1×2)	4
punto addizione, .+	195	matrice (2×1)	4
punto divisione, .P	196	matrice (2×2)	4
punto elevato a potenza, ^	197	matrice (m×n)	4
punto moltiplicazione, .*	196	prodotto (P)	5
punto sottrazione, .N	196	radice ennesima	1
riempimento, Fill	56	radice quadrata	1
scomposizione inferiore-		sistema di equazioni (2	
superiore, LU	97	equazioni)	3
scomposizione QR, QR	126	sistema di equazioni (N	
somma cumulativa,		equazioni)	3
cumulativeSum()	36	sommatoria (G)	5
somma, sum()	164	valore assoluto	4
sottomatrice, subMat()	163, 165	modi	
trasposizione, T	165	impostazione, setMode()	149
matrici casuali, randMat()	131	modulo, mod()	103
max(), massimo	98	moltiplicazione, *	192
mean(), media	98	mRow(), operazione con righe di	
media, mean()	98	matrice	103
median(), mediana	99	mRowAdd(), moltiplicazione e	
mediana, median()	99	somma di righe di matrice ..	103
MedMed, linea di regressione		MultReg (Regressione lineare	
mediana-mediana	99	multipla)	104
memorizzazione		MultRegIntervals () (Intervalli di	
simbolo, &	213-214	confidenza della previsione	
mentre, While	182	di regressione multipla) ...	104
mid(), in mezzo alla stringa	101	MultRegTests () (Verifica sulla	
min(), minimo	101	regressione lineare multipla)	105
minimo comune multiplo, lcm	101	 N	
minimo, min()	82	nand, operatore booleano	106
minore di o uguale a, {	101	nCr(), combinazioni	107
mirr(), indice interno di rendimento		nDerivative(), derivata numerica ...	108
modificato	102	newList(), nuova lista	109
mod(), modulo	103	newMat(), nuova matrice	109
modelli		nfMax(), massimo di una funzione	
derivata prima	5	calcolato numericamente ..	109
derivata seconda	6	nfMin(), minimo di una funzione	
esponente	1	calcolato numericamente ..	110
frazione	1	nInt(), integrale numerico	110
funzione esponenziale e	2	nom), converti tasso effettivo in	
funzione piecewise a 2 tratti	2	nominale	111
funzione piecewise a N tratti	3	nor, operatore boolean	111
integrale definito	6		

norm(), norma di Frobenius	112	Operatori booleani	
norma di Frobenius, norm()	112	\Rightarrow	235
norma riga matrice, rowNorm() ...	143	\Leftrightarrow	201
normale casuale, randNorm()	131	nand	106
normCdf()	112	or (Boolean), oppure	117
normPdf()	113	or, operatore booleano	117
not, operatore booleano	113	ord(), codice numerico di carattere	118
notazione in gradi, -	209	ordinamento	
notazione in gradi/primi/secondi ...	210	ascendente, SortA	158
notazione in gradienti, G	208	discendente, SortD	158
notazione in primi,	210	ottieni/restituisci	
notazione in secondi, "	210	denominatore, getDenom() ...	65
nPr(), disposizioni semplici	114	informazioni variabile,	
npv(), valore presente netto	114	getVarInfo()	69, 72
nSolve(), soluzione numerica	115	numeratore, getNum()	71
nullità, test per	81		
numerica		P	
derivata, nDeriv()	109-110	P \blacktriangleright Rx(), coordinata x rettangolare ..	118
derivata, nDerivative()	108	P \blacktriangleright Ry(), coordinata y rettangolare ..	119
integrale, nInt()	110	parte intera, iPart()	80
soluzione, nSolve()	115	parte non reale, imag()	76
nuova		passa errore, PassErr	119
lista, newList()	109	PassErr, passa errore	119
matrice, newMat()	109	Pdf()	59
O		per, For	58
oggetti		percentuale, %	197
crea collegamenti a libreria	83	piecewise()	120
OneVar, statistiche a una variabile ..	116	poissCdf()	120
operatore conversione indiretta (#) .	238	poissPdf()	121
operatore di vincolo " "	212	polare	
operatore di vincolo. ordine di		visualizza vettore, ►Polar	121
valutazione	237	polinomi	
operatori		calcola, polyEval()	122
ordine di calcolo	237	polinomi casuali, randPoly()	132
operatori boolean		polinomio casuale, randPoly()	132
and	8	polyEval(), calcola polinomio	122
nor	111	PolyRoots()	122
operatori booleani		potenza di dieci, 10^()	211
\Rightarrow	201	PowerReg, regressione su potenza ..	123
not	113	Prgm, definisci programma	124
or	117	probabilità di distribuzione normale,	
xor	183	normCdf()	112
		probabilità di distribuzione t di	
		Student, tCdf()	168

prodotto (P)			
modello di	5	radice quadrata	
prodotto vettoriale, crossP()	33	modello di	1
prodotto, product()	124	radice quadrata, $\sqrt()$	159, 203
prodotto, $\Pi()$	204	rand(), numero casuale	130
prodSeq()	124	randBin, numero casuale	130
product(), prodotto	124	randInt(), intero casuale	130
programmazione		randMat(), matrice casuale	131
definisci programma, Prgm	124	randNorm(), normale casuale	131
display data, Disp	44	randPoly(), polinomio casuale	132
passa errore, PassErr	119	randSamp()	132
visualizza dati, Vis	146	RandSeed, seme numero casuale ...	132
programmi		real(), reale	133
definizione di libreria privata ...	40	reale, real()	133
definizione di libreria pubblica ..	40	reciproco, \wedge^{-1}	212
programmi e programmazione		ref(), forma a scalini per righe	134
cancella errore, ClrErr	24	RefreshProbeVars	135
fine tentativo, EndTry	172	regressione cubica, CubicReg	35
tentativo, Try	172	regressione esponenziale, ExpReg ..	53
visualizza schermata I/O, Vis	146	regressione lineare, LinRegAx	84
visualizza schermo I/O, Disp	44	regressione lineare, LinRegBx	83, 86
propFrac, frazione propria	125	regressione logaritmica, LnReg	91
punto		regressione logistica, Logistic	94
addizione, .+	195	regressione logistica, LogisticD	95
divisione, .P	196	regressione quadratica, QuadReg ..	126
moltiplicazione, .*	196	Regressione quartica (QuartReg) ...	127
sottrazione, .N	196	regressione sinusoidale, SinReg	156
su potenza, ^	197	regressione su	
		potenza,	122-123, 136, 138,
		PowerReg ..	169
Q		regressioni	
QR, scomposizione QR	126	cubica, CubicReg	35
QuadReg, regressione quadratica ..	126	esponenziale, ExpReg	53
quando, when()	181	linea mediana-mediana, MedMed	99
QuartReg, regressione quartica	127	logaritmica, LnReg	91
R		Logistic (Regressione logistica) ..	94
R, radianti	208	logistica, Logistic	95
R▶Pr(), coordinata polare	129	MultReg (Regressione lineare multipla)	104
R▶Pθ(), coordinata polare	129	quadratica, QuadReg	126
radianti, R	208	quartica, QuartReg	127
radice ennesima		regressione lineare, LinRegAx ..	84
modello di	1	regressione lineare, LinRegBx ..	83, 86

regressione su potenza,		
PowerReg	123	
sinusoidale, SinReg	156	
regressioni su potenza, PowerReg	136, 138	
regressions, regressioni		
regressione su potenza,		
PowerReg	122, 169	
remain(), resto	136	
Request	136	
RequestStr	138	
restituisci, Return	139	
resto, remain()	136	
Return, restituisci	139	
riempi	225-226	
right(), destra	139	
right, right()	50, 181	
rimuovi		
elementi vuoti dalla lista	42	
ripeti, Cycle	36	
risultati a due variabili, TwoVar	177	
risultati, statistica	160	
risultato (ultimo), Ans	12	
rk23(), funzione Runge-Kutta	139	
rotate(), ruota	141	
round(), arrotondamento	142	
rowAdd(), somma di righe di matrice	143	
rowDim(), dimensione righe di matrice		
matrice	143	
rowNorm(), norma riga matrice	143	
rowSwap(), inverti righe matrice	144	
rref(), forma a scalini ridotta per righe	144	
ruota, rotate()	141	
S		
sblocco di variabili e gruppi di variabili		
scelte rapide della tastiera	179	
scelte rapide, tastiera	235	
scomposizione QR, QR	235	
se, If	126	
sec ⁻¹ (), secante inversa	74	
sec(), secante	145	
sech ⁻¹ (), secante iperbolica inversa	145	
sech(), secante iperbolica	146	
segno negativo, introduzione di numeri negativi	145	
segno, sign()	238	
seme numero casuale, RandSeed	152	
seno, sin()	132	
seq(), sequenza	154	
seqGen()	147	
seqn()	147	
sequence, seq()	148	
sequenza, seq()	147-148	
setMode(), imposta modo	147	
shift(), sposta	149	
sign(), segno	151	
simult(), equazioni simultanee	152	
sin ⁻¹ (), arcoseno	153	
sin(), seno	154	
sinh ⁻¹ (), arcoseno iperbolico	154	
sinh(), seno iperbolico	155	
sinistra, left()	82	
SinReg, regressione sinusoidale	156	
sistema di equazioni (2 equazioni)		
modello di	3	
sistema di equazioni (N equazioni)		
modello di	3	
somma cumulativa, cumulativeSum()		
)	36	
somma dell'interesse pagato	205	
somma di capitale versato	206	
somma di righe di matrice, rowAdd()	143	
somma, sum()	164	
somma, Σ()	205	
sommatoria (G)		
modello di	5	
SortA, ordinamento ascendente	158	
SortD, ordinamento discendente	158	
sostituzione con l'operatore " "	212	
sottomatrice, subMat()	163, 165	
sottrazione, -	191	
sposta, shift()	151	
sqrt(), radice quadrata	159	
stat.results	160	
stat.values	161	

statistica			
combinazioni, nCr()	107	strings	
deviazione standard,		right, right()	50, 181
stdDev()	161-162, 179	subMat(), sottomatrice	163, 165
disposizioni semplici, nPr()	114	sum(), somma	164
fattoriale, !	202	sumIf()	164
media, mean()	98	sumSeq()	165
mediana, median()	99		
normale casuale, randNorm()	131	T	
risultati a due variabili, TwoVar	177	T, trasposizione	165
statistiche a una variabile,		tabella di ammortamento, amortTbl(
OneVar	116)	7, 16
varianza, variance()	180	tan ⁻¹ (), arcotangente	166
statistica seme numero casuale,		tan(), tangente	166
RandSeed	132	tangente, tan()	166
statistiche a una variabile, OneVar	116	tanh ⁻¹ (), arcotangente iperbolica	168
stdDevPop(), deviazione standard		tanh(), tangente iperbolica	167
della popolazione	161	tasso di variazione media, avgRC()	15
stdDevSamp(), deviazione standard		tasso effettivo, eff()	47
del campione	162	tasso nominale, nom()	111
string(), da espressione a stringa	163	tCdf(), probabilità di distribuzione t	
stringa		di Student	168
dimensione, dim()	43	test per nullità, isVoid()	81
lunghezza	43	Test_2S, verifica F su due campioni	61
stringa di caratteri, char()	21	Text, comando	169
stringa formato, format()	58	tInterval, intervallo di confidenza t	169
stringhe		tInterval_2Samp, intervallo di	
aggiunge, &	202	confidenza t su due	
codice di carattere, ord()	118	campioni	170
conversione indiretta, #	207	tPdf(), densità di probabilità t di	
da espressione a stringa, string()	163	Student	171
da stringa a espressione, expr()	53	trace() (Traccia)	171
destra, right()	78, 139	trasposizione, T	165
dimensione, dim()	43	trattino basso, _	211
formato, format()	58	Try, comando gestione errore	172
formattazione	58	tTest, verifica t	173
in mezzo alla stringa, mid()	101	tTest_2Samp, verifica t su due	
ruota, rotate()	141	campioni	174
sinistra, left()	82	tvmFV()	175
sposta, shift()	151	tvmI()	175
stringa di caratteri, char()	21	tvmN()	175
utilizzo per creare nomi di		tvmPmt()	176
variabile	238	tvmPV()	176
TwoVar, risultati a due variabili		TwoVar, risultati a due variabili	177

U			
uguale, =	198	prodotto vettoriale, crossP()	33
unitV(), vettore unità	179	unità, unitV()	179
unLock, sblocca variabile o gruppo di variabili	179	visualizza vettore in forma cilindrica, ►Cylind	37
V		Vis, visualizza dati	146
vai a, Goto	73	visualizza come	
valore assoluto		angolo decimale, ►DD	38
modello di	4	binario, ►Base2	17
valore del denaro rapportato al tempo, importo della rata	176	esadecimale, ►Base16	19
valore del denaro rapportato al tempo, interesse	175	gradi/primi/secondi, ►DMS	46
valore del denaro rapportato al tempo, numero di rate	175	numero decimale, ►Base10	18
valore del denaro rapportato al tempo, valore futuro	175	vettore in forma cilindrica, ►Cylind	37
valore del denaro rapportato al tempo, valore presente	176	vettore polare, ►Polar	121
valore presente netto, npv()	114	vettore sferico, ►Sphere	158
valori dei risultati, statistica	161	visualizza come vettore rettangolare, ►Rect	133
variabile		visualizza dati, Disp	44
creazione di un nome da una stringa di caratteri	238	visualizza dati, Vis	146
variabile locale, Local	92	visualizza gradi/primi/secondi, ►DMS	46
variabili		visualizza vettore in forma cilindrica, ►Cylind	37
cancella tutte con il nome di un solo carattere	24	visualizza vettore sferico, ►Sphere	158
elimina, DelVar	41	W	
locale, Local	92	warnCodes(), Warning codes	181
variabili e funzioni		when(), quando	181
copia	26	While, mentre	182
variabili, blocco e sblocco	69, 93, 179	X	
varianza, variance()	180	x ² , quadrato	195
varPop()	179	XNOR	201
varSamp(), varianza campione	180	xor, or esclusivo booleano	183
verifica numero primo, isPrime()	81	Z	
Verifica t sulla regressione lineare multipla	105	zInterval, intervallo di confidenza z	184
verifica t, tTest	173	zInterval_1Prop, intervallo di confidenza z per una proporzione	184
vettore unità, unitV()	179	zInterval_2Prop, intervallo di confidenza z per due proporzioni	185
vettori		zInterval_2Samp, intervallo di	
prodotto scalare, dotP()	46		186

confidenza z su due campioni	
zTest	186
zTest_1Prop, verifica z per una proporzione	187
zTest_2Prop, verifica z per due proporzioni	188
zTest_2Samp, verifica z su due campioni	189

Δ

Δlist(), differenza in una lista	89
---	----

X

χ^2 Cdf()	22
χ^2 GOF	23
χ^2 Pdf()	24